

Alexander Gelbukh  
Ángel Fernando Kuri Morales (Eds.)

LNAI 4827

# MICAI 2007: Advances in Artificial Intelligence

6th Mexican International Conference on Artificial Intelligence  
Aguascalientes, Mexico, November 2007  
Proceedings



 Springer

Lecture Notes in Artificial Intelligence 4827

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science



Alexander Gelbukh  
Ángel Fernando Kuri Morales (Eds.)

# MICAI 2007: Advances in Artificial Intelligence

6th Mexican International Conference  
on Artificial Intelligence  
Aguascalientes, Mexico, November 4-10, 2007  
Proceedings

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA  
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Alexander Gelbukh  
Centro de Investigación en Computación, Instituto Politécnico Nacional  
Col. Nueva Industrial Vallejo, 07738, DF, Mexico  
E-mail: gelbukh@gelbukh.com

Ángel Fernando Kuri Morales  
Rio Hondo No. 1, Tizapán San Angel  
México, 01080, DF, Mexico  
E-mail: akuri@itam.mx

Library of Congress Control Number: 2007938405

CR Subject Classification (1998): I.2, F.1, I.4, F.4.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743  
ISBN-10 3-540-76630-8 Springer Berlin Heidelberg New York  
ISBN-13 978-3-540-76630-8 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media  
springer.com

© Springer-Verlag Berlin Heidelberg 2007  
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India  
Printed on acid-free paper SPIN: 12187607 06/3180 5 4 3 2 1 0

# Preface

Artificial Intelligence is a branch of computer science that studies heuristic methods of solving complex problems. Historically the first such tasks modeled human intellectual activity: reasoning, learning, seeing and speaking. Later similar methods were extended to super-complex optimization problems that appear in science, social life and industry. Many methods of Artificial Intelligence are borrowed from nature, where there occur similar super-complex problems such as those related to survival, development, and behavior of living organisms.

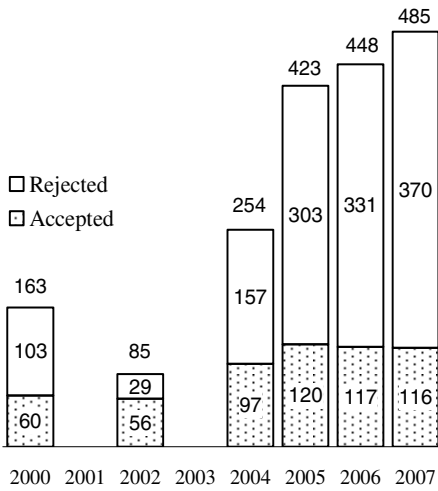
The Mexican International Conference on Artificial Intelligence (MICAI), a yearly international conference series organized by the Mexican Society for Artificial Intelligence (SMIA), is a major international AI forum and the main event in the academic life of the country's growing AI community. The proceedings of the previous MICAI events were published by Springer in its Lecture Notes in Artificial Intelligence (LNAI) series, vol. 1793, 2313, 2972, 3789, and 4293. Since its foundation in 2000, the conference has shown a stable growth in popularity (see Figures 1 and 3) and improvement in quality (see Fig. 2). The 25% acceptance rate milestone was passed for the first time this year.

This volume contains the papers presented at the oral session of the 6<sup>th</sup> Mexican International Conference on Artificial Intelligence, MICAI 2007, held on November 4–10, 2007, in Aguascalientes, Mexico. The conference received for evaluation 485 submissions by 1014 authors from 43 different countries, see Tables 1 and 2. This book contains the revised versions of 115 papers from 31 countries selected for oral presentation according to the results of the international reviewing process. Thus the acceptance rate was 23.9%. The book has been structured into 12 thematic fields representative of the main current areas of interest for the AI community:

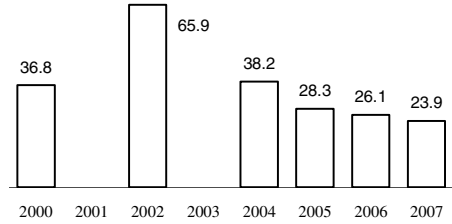
- Computational Intelligence,
- Neural Networks,
- Knowledge Representation and Reasoning,
- Agents and Multiagent Systems,
- Machine Learning and Data Mining,
- Image Processing, Computer Vision, and Robotics,
- Natural Language Processing,
- Speech Processing and Human-Computer Interfaces,
- Planning and Scheduling,
- Bioinformatics and Medical Applications,
- Industrial Applications, and
- Intelligent Tutoring Systems.

The conference featured excellent keynote lectures by leading AI experts:

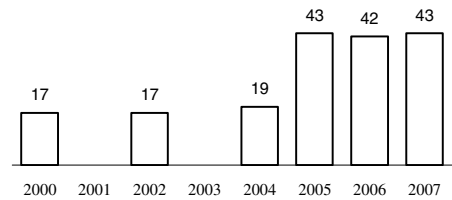
- Fernando De Arriaga-Gómez of the Polytechnic University of Madrid, Spain, who spoke about intelligent e-learning systems;



**Fig. 1.** Number of received, rejected, and accepted papers



**Fig.2.** Acceptance rate.



**Fig. 3.** Number of countries from which submissions were received

- Francisco Escolano, Director of the Robot Vision Group of the University of Alicante, Spain, who spoke about computer vision for autonomous robots and visually impaired people;
- Simon Haykin, Director of the Adaptive Systems Laboratory of the McMaster University, Canada, who spoke about neural networks, their theoretical foundations and applications;
- Pablo Noriega of the Institute for Artificial Intelligence Research of the Superior Council of Scientific Research, Spain, who spoke about regulated agent systems and automated negotiation;
- Paolo Petta of the Institute for Medical Cybernetics and Artificial Intelligence of the Centre for Brain Research of the Medical University of Vienna and Austrian Research Institute for Artificial Intelligence, Austria, who spoke about intelligent interface agents with emotions; and
- Boris Stilman of the University of Colorado at Denver, Chairman & CEO of Stilman Advanced Strategies, USA, who spoke about industrial and military applications of linguistic geometry.

In addition to the oral technical session and the keynote lectures, the conference program included tutorials (some of them given by the keynote speakers in their respective areas of expertise), workshops, and poster sessions, which were published in separate proceedings volumes and special issues of journals.

The following papers received the Best Paper Award and the Best Student Paper Award, correspondingly (the best student paper was selected out of papers whose first author was a full-time student):

1<sup>st</sup> place: *Scaling Kernels: A New Least Squares Support Vector Machine Kernel for Approximation*, by Mu Xiangyang, Zhang Taiyi and Zhou Yatong (China);

**Table 1.** Statistics of submissions and accepted papers by country / region

Country / Region	Authors		Papers		Country / Region	Authors		Papers	
	Submitted	Accepted	Accepted	Rate		Submitted	Accepted	Accepted	Rate
Algeria	3	2.33	1	0.43	Japan	12	5.75	3	0.52
Argentina	32	12.42	2.42	0.19	Korea, South	30	16.33	1.5	0.09
Australia	4	2	0	0	Macau	1	0.5	0	0
Austria	3	1.5	0.5	0.33	Macedonia	1	0.5	0	0
Belgium	3	1.17	0.5	0.43	Mexico	409	194.2	42.67	0.22
Brazil	37	20	6.67	0.33	Pakistan	6	3	0	0
Canada	5	1.5	0	0	Peru	1	0.5	0	0
Chile	20	15.67	1.5	0.1	Poland	4	3	2	0.67
China	124	53.17	7.75	0.15	Portugal	13	4.5	0.5	0.11
Colombia	21	8.47	1.67	0.2	Romania	2	1	0	0
Cuba	22	9.75	2.5	0.26	Russia	8	3.08	0.33	0.11
Czech Rep.	8	2.67	1.67	0.62	Spain	75	28.67	10.83	0.38
Egypt	1	1	0	0	Sweden	3	2	1	0.50
Finland	2	1	0	0	Switzerland	5	2.58	2.33	0.90
France	35	15.89	6.31	0.4	Taiwan	19	12	1	0.08
Germany	10	6.86	4.86	0.71	Thailand	2	1	1	1
Greece	4	2	0	0	Tunisia	5	1.75	0.75	0.43
Hong Kong	1	0.25	0.25	1	Turkey	16	8	1	0.13
India	3	1	0	0	UAE	1	1	0	0
Iran	21	17	3	0.18	UK	8	4.33	2	0.46
Israel	1	0.25	0.25	1	USA	26	12.17	4.25	0.35
Italy	7	3.25	1	0.31	Total:	1014	485	116	

<sup>1</sup> Counted by authors: e.g., for a paper by 2 authors from UK and 1 from USA, we added  $\frac{2}{3}$  to UK and  $\frac{1}{3}$  to USA.

2<sup>nd</sup> place: *On-line Rectification of Sport Sequences with Moving Cameras*, by Jean-Bernard Hayet and Justus Piater (Mexico / Belgium);

3<sup>rd</sup> place: *SELDI-TOF-MS Pattern Analysis for Cancer Detection as a Base for Diagnostic Software*, by Marcin Radlak and Ryszard Klempous (UK / Poland);

Student: *3D Object Recognition Based on Low Frequency Response and Random Feature Selection*, by Roberto A. Vázquez, Humberto Sossa and Beatriz A. Garro (Mexico).

We want to thank all those involved in the organization of this conference. In the first place, these are the authors of the papers constituting this book: it is the excellence of their research work that gives value to the book and sense to the work of all of the other people involved. We thank the members of the Program Committee and additional reviewers for their great and very professional work on reviewing and selecting the papers for the conference. Our very special thanks go to the members of the Board of Directors of SMIA, especially to José Galaviz Casas, Sulema Torres, Yulia Ledeneva, and Alejandro Peña of the CIC-IPN and Oscar Celma of the Music Technology Group of the Pompeu Fabra University devoted great effort to the preparation of the conference. Mikhail Alexandrov, Hiram Calvo, Denis Filatov,

**Table 2.** Statistics of submissions and accepted papers by topic<sup>2</sup>

Accepted	Submitted	Rate	Topic
32	96	0.33	Machine Learning
21	72	0.29	Neural Networks
19	97	0.20	Other
19	67	0.28	Natural Language Processing and Understanding
19	55	0.35	Computer Vision
14	52	0.27	Knowledge Representation
14	35	0.40	Hybrid Intelligent Systems
13	50	0.26	Data Mining
13	49	0.27	Genetic Algorithms
11	40	0.28	Planning and Scheduling
10	51	0.20	Fuzzy Logic
8	29	0.28	Knowledge Management
7	40	0.17	Robotics
7	27	0.26	Uncertainty and Probabilistic Reasoning
7	26	0.27	Knowledge Acquisition
7	20	0.35	Constraint Programming
7	16	0.44	Logic Programming
6	26	0.23	Bioinformatics
6	21	0.29	Intelligent Interfaces: Multimedia; Virtual Reality
5	28	0.18	Expert Systems and Knowledge-Based Systems
5	17	0.29	Computational Creativity
4	32	0.12	Multiagent systems and Distributed AI
4	10	0.40	Model-Based Reasoning
4	8	0.50	Belief Revision
3	13	0.23	Navigation
3	10	0.30	Nonmonotonic Reasoning
3	9	0.33	Spatial and Temporal Reasoning
3	7	0.43	Qualitative Reasoning
2	6	0.33	Intelligent Organizations
2	5	0.40	Common Sense Reasoning
1	21	0.05	Ontologies
1	16	0.06	Intelligent Tutoring Systems
1	9	0.11	Case-Based Reasoning
1	5	0.20	Philosophical and Methodological Issues of AI
1	3	0.33	Automated Theorem Proving
0	10	0.00	Knowledge Verification; Sharing; Reuse
0	4	0.00	Assembly

<sup>2</sup> According to the topics indicated by the authors. A paper may have more than one topic.

Oleksiy Pogrebnyak, Grigory Sidorov, Manuel Vilares were among the most helpful and active PC members.

We would like to express our sincere gratitude to the IEEE Section of Aguascalientes, the Instituto Tecnológico de Aguascalientes, the Universidad Autónoma de Aguascalientes, the Universidad Tecnológica de Aguascalientes, the Tecnológico de Monterrey Campus Aguascalientes, the Universidad Politécnica de Aguascalientes and the Museo Descubre for their warm hospitality and for providing the infrastructure for the tutorials and workshops. Special thanks to the Constitutional

Governor of the State of Aguascalientes, Ing. Luis Armando Reynoso Femat, for his valuable participation and support in the organization of this conference. The opening ceremony and keynote lectures were held in the beautiful Teatro Aguascalientes which would not have been available without his decided support. We also thank the Consejo de Ciencia y Tecnología of the State of Aguascalientes for their partial financial support, and the Secretaría de Desarrollo Económico, Subsecretaría de Gestión e Innovación and Secretaría de Turismo of the State of Aguascalientes for their effort in organizing industrial and tourist visits as well as cultural and amusement activities. We are deeply grateful to the conference staff and to all of the members of the Local Committee headed by José Antonio Calderón Martínez.

The entire submission and reviewing process, as well as the assemblage of the proceedings, was freely supported by the EasyChair system ([www.easychair.org](http://www.easychair.org)); we express our gratitude to its author, Andrei Voronkov, for his constant support and help. Last but not least, we deeply appreciate the patience of the staff at Springer and their help in editing this volume.

September 2007

Alexander Gelbukh  
Angel Kuri

# Organization

MICAI 2007 was organized by the Mexican Society for Artificial Intelligence (SMIA) in collaboration with the IEEE Section of Aguascalientes, the Instituto Tecnológico de Aguascalientes, the Universidad Autónoma de Aguascalientes, the Universidad Tecnológica de Aguascalientes, the Tecnológico de Monterrey Campus Aguascalientes, the Universidad Politécnica de Aguascalientes and the Museo Descubre, as well as the Center for Computing Research of the National Polytechnic Institute (CIC-IPN), the Instituto Tecnológico Autónomo de México (ITAM), the Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE), the Universidad Nacional Autónoma de México (UNAM), the Universidad Autónoma de México (UAM-Azcapotzalco), and the Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM), Mexico.

## Conference Committee

General Chairs	Ángel Kuri Morales Carlos Alberto Reyes-Garcia
Program Chairs	Alexander Gelbukh Ángel Kuri Morales
Workshop and Tutorial Chair	Raul Monroy
Student Chair	José Galaviz Casas
Finance Chair	Ana Lilia Laureano Cruces
Award Selection Committee	Ángel Kuri Morales Alexander Gelbukh

## Program Committee

Ajith Abraham	Louise Dennis
Mikhail Alexandrov	Juergen Dix
Gustavo Arroyo	Abdenmour El Rhalibi
Ildar Batyrshin	Denis Filatov
Bedrich Benes	José Galaviz
Igor Bolshakov	Sofía N. Galicia-Haro
Paul Brna	Matjaž Gams
Andre C. P. L. F. de Carvalho	Alexander Gelbukh (Co-chair)
Hiram Calvo	Arturo Hernández-Aguirre
Nicoletta Calzolari	Jesse Hoey
María José Castro Bleda	Dieter Hutter
Simon Colton	Pablo H. Ibarguengoytia
Ulises Cortes	Ryszard Klempons
Nareli Cruz-Cortes	Mario Köppen



Angel Kuri (Co-chair)  
Ana Lilia Laureano-Cruces  
Steve Legrand  
Christian Lemaître León  
Eugene Levner  
James Little  
Aurelio López  
Jacek Malec  
Efren Mezura-Montes  
Mikhail Mikhailov  
Chilukuri Mohan  
Raúl Monroy  
Eduardo Morales  
Guillermo Morales Luna  
Juan Arturo Nolazco Flores  
Mauricio Osorio Galindo  
Manuel Palomar  
Oleksiy Pogrebnyak  
Fuji Ren  
Carlos Alberto Reyes-García  
Riccardo Rosati

Paolo Rosso  
Stefano Rovetta  
Khalid Saeed  
Andrea Schaerf  
Leonid Sheremetov  
Grigori Sidorov  
Humberto Sossa Azuela  
Benno Stein  
Thomas Stuetzle  
Luis Sucar  
Hugo Terashima  
Berend Jan van der Zwaag  
Javier Vázquez-Salceda  
Manuel Vilares Ferro  
Luis Villaseñor-Pineda  
Toby Walsh  
Alfredo Weitzenfeld  
Franz Wotawa  
Kaori Yoshida  
Carlos Mario Zapata Jaramillo

### **Additional Referees**

Mohamed Abdel Fattah  
Omar Abuelma'atti  
Luis Alberto Pineda  
Jose Arrazola  
Héctor Avilés  
Alejandra Barrera  
Lucia Barron  
Marcio Basgalupp  
Mike Baskett  
Tristan Behrens  
Edgard Benítez-Guerrero  
Edmundo Bonilla  
Matthew Brisbin  
Nils Bulling  
Sara Carrera Carrera  
Oscar Celma  
Zenon Chaczko  
Marco Chiarandini  
Victor Manuel Darriba Bilbao  
Rogelio Davila Perez  
Yogesh Deshpande  
Joseph Devereux  
Luca Di Gaspero

Pantelis Elinas  
Cora Beatriz Excelente Toledo  
Katti Faceli  
Bruno Feres de Souza  
Milagros Fernández-Gavilanes  
Michael Francis  
Peter Funk  
Karen Azurim Garcia Gamboa  
René A. García Hernández  
Paola Garcia-Perera  
Sergio Gomez  
Fabio A. Gonzalez  
John Haggerty  
Emmanuel Hebrard  
Abir Hussain  
Diana Inkpen  
Ruben Izquierdo-Bevia  
Bartosz Jablonski  
Peilin Jiang  
Phil Kilby  
Jerzy Kotowski  
Zornitsa Kozareva  
Ricardo Landa Becerra

Yulia Ledeneva	David Pinto
Domenico Lembo	Antonella Poggi
Agustin Leon	Martin Potthast
Erica Lloves Calviño	Pascal Poupart
Juan Carlos Lopez Pimentel	Pilar Pozos
Jose Luis Carballido	Jakob Puchinger
Michael Maher	Orion Fausto Reyes-Galaviz
Antonio Marin Hernandez	Francisco Ribadas-Pena
Manuel Mejia	Andrea Roli
Carlos Mex-Perera	Israel Román
Sven Meyer zu Eissen	Marco Ruzzi
Rada Mihalcea	William Sellers
Miguel Angel Molinero Alvarez	Eduardo Spinosa
Manuel Montes-y-Gómez	Geof Staniford
Rafael Morales	Gerald Steinbauer
Paloma Moreda Pozo	Ewa Szlachcic
Boris Motik	Sulema Torres
Rafael Murrieta	Gregorio Toscano-Pulido
Mariá Nascimento	Victor Trevino
Juan Antonio Navarro	Dan Tufis
Juan Carlos Nieves	Javier Vazquez
Peter Novak	Andrew Verden
Slawomir Nowaczyk	Joerg Weber
Constantin Orasan	Ning Xiong
Ryan Pedela	Fernando Zacarias-Flores
Bernhard Peischl	Ramon Zatarain
Alejandro Peña Ayala	Claudia Zepeda Cortes

## Organizing Committee

Chair	Jose Antonio Calderon Martínez (IEEE Aguascalientes Section Chair, ITA)
Logistics	Luis Enrique Arambula Miranda (UAA) Eduardo Lopez Guzman (Descubre) Juan Carlos Lira Padilla (Descubre)
Publicity	Juan Manuel Campos Sandoval (ITESM-CA) Alejandro Davila Viramontes (UPA)
Finance and Sponsorship	Raúl Gutierrez Perucho (ITESM-CA) Victor Manuel Gonzalez Arredondo (UTA)

## Webpage and Contact

The MICAI series webpage is at [www.MICAI.org](http://www.MICAI.org). The webpage of the Mexican Society for Artificial Intelligence, SMIA, is at [www.SMIA.org.mx](http://www.SMIA.org.mx). Contact options and additional information can be found on those webpages.

# Table of Contents

## Computational Intelligence

Rough Set Approach Under Dynamic Granulation in Incomplete Information Systems .....	1
Generalized Fuzzy Operations for Digital Hardware Implementation ....	9
A Novel Model of Artificial Immune System for Solving Constrained Optimization Problems with Dynamic Tolerance Factor .....	19
A Genetic Representation for Dynamic System Qualitative Models on Genetic Programming: A Gene Expression Programming Approach .....	30
Handling Constraints in Particle Swarm Optimization Using a Small Population Size .....	41
Collective Methods on Flock Traffic Navigation Based on Negotiation ...	52
A New Global Optimization Algorithm Inspired by Parliamentary Political Competitions .....	61
Discovering Promising Regions to Help Global Numerical Optimization Algorithms .....	72
Clustering Search Approach for the Traveling Tournament Problem ....	83
Stationary Fokker – Planck Learning for the Optimization of Parameters in Nonlinear Models .....	94
From Horn Strong Backdoor Sets to Ordered Strong Backdoor Sets ....	105

G-Indicator: An M-Ary Quality Indicator for the Evaluation of Non-dominated Sets ..... 118

Approximating the  $\epsilon$ -Efficient Set of an MOP with Stochastic Search Algorithms ..... 128

A Multicriterion SDSS for the Space Process Control: Towards a Hybrid Approach ..... 139

**Neural Networks**

Radial Basis Function Neural Network Based on Order Statistics ..... 150

Temperature Cycling on Simulated Annealing for Neural Network Learning ..... 161

On Conditions for Intermittent Search in Self-organizing Neural Networks ..... 172

Similarity Clustering of Music Files According to User Preference ..... 182

Complete Recall on Alpha-Beta Heteroassociative Memory ..... 193

**Knowledge Representation and Reasoning**

I-Cog: A Computational Framework for Integrated Cognition of Higher Cognitive Abilities ..... 203

A Rule-Based System for Assessing Consistency Between UML Models ..... 215

Partial Satisfiability-Based Merging ..... 225

Optimizing Inference in Bayesian Networks and Semiring Valuation Algebras .....	236
Compiling Solution Configurations in Semiring Valuation Systems .....	248
Implementing Knowledge Update Sequences .....	260
On Reachability of Minimal Models of Multilattice-Based Logic Programs .....	271
Update Sequences Based on Minimal Generalized Pstable Models .....	283
PStable Semantics for Possibilistic Logic Programs .....	294
Improving Efficiency of Prolog Programs by Fully Automated Unfold/Fold Transformation .....	305
A Word Equation Solver Based on Levensthein Distance .....	316
Simple Model-Based Exploration and Exploitation of Markov Decision Processes Using the Elimination Algorithm .....	327
A Simple Model for Assessing Output Uncertainty in Stochastic Simulation Systems .....	337

## Agents and Multiagent Systems

An Empirically Terminological Point of View on Agentism in the Artificial .....	348
Inductive Logic Programming Algorithm for Estimating Quality of Partial Plans .....	359
Modeling Emotion-Influenced Social Behavior for Intelligent Virtual Agents .....	370

Just-in-Time Monitoring of Project Activities Through Temporal Reasoning ..... 381

**Machine Learning and Data Mining**

Scaling Kernels: A New Least Squares Support Vector Machine Kernel for Approximation ..... 392

Evolutionary Feature and Parameter Selection in Support Vector Regression ..... 399

Learning Models of Relational MDPs Using Graph Kernels ..... 409

Weighted Instance-Based Learning Using Representative Intervals ..... 420

A Novel Information Theory Method for Filter Feature Selection ..... 431

Building Fine Bayesian Networks Aided by PSO-Based Feature Selection ..... 441

Two Simple and Effective Feature Selection Methods for Continuous Attributes with Discrete Multi-class ..... 452

INCRAIN: An Incremental Approach for the Gravitational Clustering ..... 462

On the Influence of Class Information in the Two-Stage Clustering of a Human Brain Tumour Dataset ..... 472

Learning Collaboration Links in a Collaborative Fuzzy Clustering Environment ..... 483

Algorithm for Graphical Bayesian Modeling Based on Multiple Regressions ..... 496

Coordinating Returns Policies and Marketing Plans for Profit Optimization in E-Business Based on a Hybrid Data Mining Process ...	507
An EM Algorithm to Learn Sequences in the Wavelet Domain .....	518
Assessment of Personal Importance Based on Social Networks .....	529
Optimization Procedure for Predicting Nonlinear Time Series Based on a Non-Gaussian Noise Model .....	540
An Improved Training Algorithm of Neural Networks for Time Series Forecasting .....	550
Evolved Kernel Method for Time Series .....	559
 <b>Image Processing, Computer Vision, and Robotics</b>	
Using Ant Colony Optimization and Self-organizing Map for Image Segmentation .....	570
Correspondence Regions and Structured Images .....	580
The Wavelet Based Contourlet Transform and Its Application to Feature Preserving Image Coding .....	590
Design of an Evolutionary Codebook Based on Morphological Associative Memories .....	601
A Shape-Based Model for Visual Information Retrieval .....	612
An Indexing and Retrieval System of Historic Art Images Based on Fuzzy Shape Similarity .....	623
PCB Inspection Using Image Processing and Wavelet Transform .....	634

A Single-Frame Super-Resolution Innovative Approach ..... 640  
.....

Shadows Attenuation for Robust Object Recognition ..... 650  
.....

Fuzzy Directional Adaptive Recursive Temporal Filter for Denoising of  
Video Sequences ..... 660  
.....

Bars Problem Solving - New Neural Network Method and  
Comparison ..... 671  
.....

A Coarse-and-Fine Bayesian Belief Propagation for Correspondence  
Problems in Computer Vision ..... 683  
.....

3D Object Recognition Based on Low Frequency Response and Random  
Feature Selection ..... 694  
.....

Image Processing for 3D Reconstruction Using a Modified Fourier  
Transform Profilometry Method ..... 705  
.....

3D Space Representation by Evolutive Algorithms ..... 713  
.....

Knowledge Acquisition and Automatic Generation of Rules for the  
Inference Machine CLIPS ..... 725  
.....

On-Line Rectification of Sport Sequences with Moving Cameras..... 736  
.....

A New Person Tracking Method for Human-Robot Interaction Intended  
for Mobile Devices ..... 747  
.....



Example-Based Face Shape Recovery Using the Zenith Angle of the Surface Normal .....	758
Feature Extraction and Face Verification Using Gabor and Gaussian Mixture Models .....	769
Lips Shape Extraction Via Active Shape Model and Local Binary Pattern .....	779
Continuous Stereo Gesture Recognition with Multi-layered Silhouette Templates and Support Vector Machines .....	789
Small-Time Local Controllability of a Differential Drive Robot with a Limited Sensor for Landmark-Based Navigation .....	800
<b>Natural Language Processing</b>	
Learning Performance in Evolutionary Behavior Based Mobile Robot Navigation .....	811
Fuzzifying Clustering Algorithms: The Case Study of MajorClust .....	821
Taking Advantage of the Web for Text Classification with Imbalanced Classes .....	831
A Classifier System for Author Recognition Using Synonym-Based Features .....	839
Variants of Tree Kernels for XML Documents .....	850
Textual Energy of Associative Memories: Performant Applications of Enertex Algorithm in Text Summarization and Topic Segmentation .....	861

A New Hybrid Summarizer Based on Vector Space Model, Statistical Physics and Linguistics .....	872
Graph Decomposition Approaches for Terminology Graphs .....	883
An Improved Fast Algorithm of Frequent String Extracting with no Thesaurus .....	894
Using Lexical Patterns for Extracting Hyponyms from the Web .....	904
On the Usage of Morphological Tags for Grammar Induction .....	912
Web-Based Model for Disambiguation of Prepositional Phrase Usage ...	922
Identification of Chinese Verb Nominalization Using Support Vector Machine .....	933
Enrichment of Automatically Generated Texts Using Metaphor .....	944
An Integrated Reordering Model for Statistical Machine Translation....	955
Hobbs' Algorithm for Pronoun Resolution in Portuguese .....	966
Automatic Acquisition of Attribute Host by Selectional Constraint Resolution .....	975
E-Gen: Automatic Job Offer Processing System for Human Resources .....	985
How Context and Semantic Information Can Help a Machine Learning System? .....	996

## Speech Processing and Human-Computer Interfaces

Auditory Cortical Representations of Speech Signals for Phoneme Classification .....	1004
Using Adaptive Filter and Wavelets to Increase Automatic Speech Recognition Rate in Noisy Environment .....	1015
Spoken Commands in a Smart Home: An Iterative Approach to the Sphinx Algorithm .....	1025
Emotion Estimation Algorithm Based on Interpersonal Emotion Included in Emotional Dialogue Sentences .....	1035
The Framework of Mental State Transition Analysis .....	1046

## Planning and Scheduling

Integration of Symmetry and Macro-operators in Planning .....	1056
Planning by Guided Hill-Climbing .....	1067
DiPro: An Algorithm for the Packing in Product Transportation Problems with Multiple Loading and Routing Variants .....	1078
On the Performance of Deterministic Sampling in Probabilistic Roadmap Planning .....	1089
Hybrid Evolutionary Algorithm for Flowtime Minimisation in No-Wait Flowshop Scheduling .....	1099
Enhancing Supply Chain Decisions Using Constraint Programming: A Case Study .....	1110

## Bioinformatics and Medical Applications

Analysis of DNA-Dimer Distribution in Retroviral Genomes Using a Bayesian Networks Induction Technique Based on Genetic Algorithms ..... 1122

SELDI-TOF-MS Pattern Analysis for Cancer Detection as a Base for Diagnostic Software ..... 1132

Three Dimensional Modeling of Individual Vessels Based on Matching of Adaptive Control Points ..... 1143

## Industrial Applications

Design and Implementation of Petrinet Based Distributed Control Architecture for Robotic Manufacturing Systems ..... 1151

Multi Sensor Data Fusion for High Speed Machining ..... 1162

VisualBlock-FIR for Fault Detection and Identification: Application to the DAMADICS Benchmark Problem ..... 1173

Sliding Mode Control of a Hydrocarbon Degradation in Biopile System Using Recurrent Neural Network Model ..... 1184

## Intelligent Tutoring Systems

Knowledge Acquisition in Intelligent Tutoring System: A Data Mining Approach ..... 1195

Features Selection Through FS-Testors in Case-Based Systems of Teaching-Learning ..... 1206

Heuristic Optimization Methods for Generating Test from a Question Bank ..... 1218

**Author Index** ..... 1231

# Rough Set Approach Under Dynamic Granulation in Incomplete Information Systems

Yuhua Qian<sup>1,2,3</sup>, Jiye Liang<sup>1,2</sup>, Xia Zhang<sup>1,2</sup>, and Chuangyin Dang<sup>3</sup>

<sup>1</sup> School of Computer and Information Technology, Shanxi University  
Taiyuan, 030006, Shanxi, China

<sup>2</sup> Key Laboratory of Computational Intelligence and Chinese Information Processing  
of Ministry of Education, Taiyuan, 030006, Shanxi, China

<sup>3</sup> Department of Manufacturing Engineering and Engineering Management, City  
University of Hong Kong, Hong Kong

jinchengqyh@126.com, ljiy@sxu.edu.cn, zhangxia@sxu.edu.cn,  
mecdang@cityu.edu.hk

**Abstract.** In this paper, the concept of a granulation order is proposed in an incomplete information system. Positive approximation of a set under a granulation order is defined and its some useful properties are investigated. Unlike classical rough set, this approach focuses on how to describe the structure of a rough set in incomplete information systems. For a subset of the universe, its approximation accuracy is monotonously increasing under a granulation order. This means that a proper family of granulations can be chosen for a target-concept approximation according to user requirements.

**Keywords:** Information systems, granular computing, dynamic granulation, partial relation.

## 1 Introduction

Granular computing is a new active area of current research in artificial intelligence, and is a new concept and computing formula for information processing. It has been widely applied to many branches of artificial intelligence such as problem solving, knowledge discovery, image processing, semantic Web services.

In 1979, the problem of fuzzy information granules was introduced by L.A. Zadeh in [1]. Then, in [2-4] he introduced a concept of granular computing, as a term with many meanings, covering all the research of theory, methods, techniques and tools related to granulation. A general model based on fuzzy set theory was proposed, and granules were defined and constructed basing on the concept of generalized constraints in [3]. Relationships among granules were represented in terms of fuzzy graphs or fuzzy if-then rules. Z. Pawlak [5] proposed that each equivalence class may be viewed as a granule consisting of indistinguishable elements, also referred to as to an equivalence granule. Some basic problems and methods such as logic framework, concept approximation, and consistent classification for granular computing were outlined by Y.Y. Yao in

[6]. The structure, modeling, and applications of granular computing under some binary relations were discussed, and the granular computing methods based on fuzzy sets and rough sets were proposed by T.Y. Lin in [7]. Quotient space theory was extended to fuzzy quotient space theory based on fuzzy equivalence relation by L. Zhang and B. Zhang in [8], providing a powerful mathematical model and tools for granular computing. By using similarity between granules, some basic issues on granular computing were discussed by G.J. Klir in [9]. Several measures in information systems closely associated with granular computing, such as granulation measure, information and rough entropy, as well as knowledge granulation, were discussed by J.Y. Liang in [10, 11]. Decision rule granules and a granular language for logical reasoning based on rough set theory were studied by Q. Liu in [12].

In the view of granular computing, a concept described by a set is always characterized via the so-called upper and lower approximations under static granulation in rough set theory, and a static boundary region of the concept is induced by the upper and lower approximations. However a concept described by using the positive approximation is characterized via the variational upper and lower approximations under dynamic granulation, which is an aspect of people's comprehensive solving ability at some different granulation spaces [13]. The positive approximation extends classical rough set, and enriches rough set theory and its application. This paper aims to extend this approach to the rough set approximation under dynamic granulation in incomplete information systems.

## 2 Positive Approximation in Incomplete Information Systems

In this section, we review some basic concepts such as incomplete information systems, tolerance relation and partial relation of knowledge, introduce the notion of positive approximation to describe the structure of a set approximation in incomplete information systems and investigate its some useful properties.

An information system is a pair  $S = (U, A)$ , where,

- (1)  $U$  is a non-empty finite set of objects;
- (2)  $A$  is a non-empty finite set of attributes;
- (3) for every  $a \in A$ , there is a mapping  $a, a : U \rightarrow V_a$ , where  $V_a$  is called the value set of  $a$ .

It may happen that some of the attribute values for an object are missing. For example, in medical information systems there may exist a group of patients for which it is impossible to perform all the required tests. These missing values can be represented by the set of all possible values for the attribute or equivalence by the domain of the attribute. To indicate such a situation, a distinguished value, a so-called null value is usually assigned to those attributes. If  $V_a$  contains a null value for at least one attribute  $a \in A$ , then  $S$  is called an incomplete information system, otherwise it is complete [14, 15]. Further on, we will denote the null value by  $*$ .

Let  $S = (U, A)$  be an information system,  $P \subseteq A$  an attribute set. We define a binary relation on  $U$  as follows

$$SIM(P) = \{(u, v) \in U \times U \mid \forall a \in P, a(u) = a(v) \text{ or } a(u) = * \text{ or } a(v) = *\}.$$

In fact,  $SIM(P)$  is a tolerance relation on  $U$ , the concept of a tolerance relation has a wide variety of applications in classification [16, 17]. It can be easily shown that  $SIM(P) = \bigcap_{a \in P} SIM(\{a\})$ .

Let  $S_P(u)$  denote the set  $\{v \in U \mid (u, v) \in SIM(P)\}$ .  $S_P(u)$  is the maximal set of objects which are possibly indistinguishable by  $P$  with  $u$ . Let  $U/SIM(P)$  denote the family sets  $\{S_P(u) \mid u \in U\}$ , the classification or the knowledge induced by  $P$ . A member  $S_P(u)$  from  $U/SIM(P)$  will be called a tolerance class or a granule of information. It should be noticed that the tolerance classes in  $U/SIM(P)$  do not constitute a partition of  $U$  in general. They constitute a cover of  $U$ , i.e.,  $S_P(u) \neq \emptyset$  for every  $u \in U$ , and  $\bigcup_{u \in U} S_P(u) = U$ .

Let  $S = (U, A)$  be an incomplete information system, we define a partial relation  $\preceq$  (or  $\succeq$ ) on  $2^A$  as follows: we say that  $Q$  is coarser than  $P$  (or  $P$  is finer than  $Q$ ), denoted by  $P \preceq Q$  (or  $Q \succeq P$ ), if and only if  $S_P(u_i) \subseteq S_Q(u_i)$  for  $i \in \{1, 2, \dots, |U|\}$ . If  $P \preceq Q$  and  $P \neq Q$ , we say that  $Q$  is strictly coarser than  $P$  (or  $P$  is strictly finer than  $Q$ ) and denoted by  $P \prec Q$  (or  $Q \succ P$ ).

In fact,  $P \prec Q \Leftrightarrow$  for  $i \in \{1, 2, \dots, |U|\}$ , we have that  $S_P(u_i) \subseteq S_Q(u_i)$ , and  $\exists j \in \{1, 2, \dots, |U|\}$ , such that  $S_P(u_j) \subset S_Q(u_j)$ .

Let  $S = (U, A)$  be an incomplete information system,  $X$  a subset of  $U$  and  $P \subseteq A$  an attribute set. In the rough set model based on tolerance relation [14],  $X$  is characterized by  $SIM(P)(X)$  and  $\overline{SIM(P)}(X)$ , where

$$\overline{SIM(P)}(X) = \bigcup \{Y \in U/SIM(P) \mid Y \subseteq X\}, \tag{1}$$

$$SIM(P)(X) = \bigcup \{Y \in U/SIM(P) \mid Y \cap X \neq \emptyset\}. \tag{2}$$

In an incomplete information system, a cover  $U/SIM(P)$  of  $U$  induced by the tolerance relation  $SIM(P)$ ,  $P \in 2^A$ , provides a granulation world for describing a concept  $X$ . So a sequence of attribute sets  $P_i \in 2^A$  ( $i = 1, 2, \dots, n$ ) with  $P_1 \succeq P_2 \succeq \dots \succeq P_n$  can determine a sequence of granulation worlds, from the most rough one to the most fine one. We define the upper and lower approximations of a concept under a granulation order.

**Definition 1.** Let  $S = (U, A)$  be an incomplete information system,  $X \subseteq U$ ,  $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$  be a sequence of attribute sets with  $P_1 \succeq P_2 \succeq \dots \succeq P_n$ ,  $P_i \in 2^A$ ,  $f_i = \overline{SIM(P_i)}(X)$ ,  $\mathbf{P}X = \bigcap_{i=1}^n \overline{SIM(P_i)}(X)$ ,  $\underline{\mathbf{P}}X = \bigcup_{i=1}^n SIM(P_i)(X)$ .

$$\overline{\mathbf{P}}(X) = \overline{SIM(P_n)}(X), \tag{3}$$

$$\underline{\mathbf{P}}(X) = \bigcup_{i=1}^n \overline{SIM(P_i)}(X_i), \tag{4}$$

$$X_1 = X, \dots, X_i = X - \bigcup_{k=1}^{i-1} \overline{SIM(P_k)}(X_k), \quad i = 2, \dots, n$$

$bn_{\mathbf{P}}(X) = \overline{\mathbf{P}}(X) - \underline{\mathbf{P}}(X)$  is called  $\mathbf{P}$ -boundary region of  $X$ ,  $pos_{\mathbf{P}}(X) = \underline{\mathbf{P}}(X)$  is called  $\mathbf{P}$ -positive region of  $X$ , and  $neg_{\mathbf{P}}(X) = U - \overline{\mathbf{P}}(X)$  is called  $\mathbf{P}$ -negative region of  $X$ . Obviously, we have  $\overline{\mathbf{P}}(X) = pos_{\mathbf{P}}(X) \cup bn_{\mathbf{P}}(X)$ .

Definition 1 shows that a target concept is approached by the change of the lower approximation  $\underline{\mathbf{P}}(X)$  and the upper approximation  $\overline{\mathbf{P}}(X)$ .

From this definition, we have the following theorem.

**Theorem 1.** . . .  $S = (U, A)$  . . . . .  $X$  . . . . .  
 $U$  . . .  $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$  . . . . .  $P_1 \succeq P_2 \succeq \dots \succeq P_n$   
 $P_i \in 2^A$  . . .  $\mathbf{P}_i = \{P_1, P_2, \dots, P_i\}$  . . . . .  $\mathbf{P}_i \ i = 1, 2, \dots, n$  . . . . .

$$\underline{\mathbf{P}}_i(X) \subseteq X \subseteq \overline{\mathbf{P}}_i(X), \quad (5)$$

$$\underline{\mathbf{P}}_1(X) \subseteq \underline{\mathbf{P}}_2(X) \subseteq \dots \subseteq \underline{\mathbf{P}}_n(X). \quad (6)$$

Theorem 1 states that the lower approximation enlarges as the granulation order become longer through adding attribute subsets, which help to describe exactly a target concept.

In [18], the approximation measure  $\alpha_R(X)$  was originally introduced by Z. Pawlak for classical lower and upper approximation, where  $\alpha_R(X) = \frac{|RX|}{|R|} (X \neq \emptyset)$ . Here we introduce the concept to the positive approximation in order to describe the uncertainty of a target concept under a granulation order.

**Definition 2.** . . .  $S = (U, A)$  . . . . .  $X$  . . . . .  
 $U$  . . .  $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$  . . . . .  $P_1 \succeq P_2 \succeq \dots \succeq P_n$   
 $P_i \in 2^A$  . . . . .  $\alpha_{\mathbf{P}}(X)$  . . . . .

$$\alpha_{\mathbf{P}}(X) = \frac{|\underline{\mathbf{P}}(X)|}{|\overline{\mathbf{P}}(X)|}, \quad (7)$$

. . .  $X \neq \emptyset$

**Theorem 2.** . . .  $S = (U, A)$  . . . . .  $X$  . . . . .  
 $U$  . . .  $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$  . . . . .  $P_1 \succeq P_2 \succeq \dots \succeq P_n$   
 $P_i \in 2^A$  . . .  $\mathbf{P}_i = \{P_1, P_2, \dots, P_i\}$  . . . . .

$$\alpha_{\mathbf{P}_1}(X) \leq \alpha_{\mathbf{P}_2}(X) \leq \dots \leq \alpha_{\mathbf{P}_n}(X). \quad (8)$$

Theorem 2 states that the approximation measure  $\alpha_{\mathbf{P}}(X)$  increases as the granulation order become longer through adding attribute subsets.

In order to illustrate the essence that the positive approximation is mainly concentrated on the change of the construction of the target concept  $X$  (tolerance classes in lower approximation of  $X$  with respect to  $\mathbf{P}$ ) in incomplete information systems, we can re-define  $\mathbf{P}$ -positive approximation of  $X$  by using some tolerance classes on  $U$ .

Therefore, the structure of  $\mathbf{P}$ -upper approximation  $\overline{\mathbf{P}}(X)$  and  $\mathbf{P}$ -lower approximation  $\underline{\mathbf{P}}(X)$  of  $\mathbf{P}$ -positive approximation of  $X$  can be represented as

$$[\overline{\mathbf{P}}(X)] = \{S_{P_n}(u) \mid S_{P_n}(u) \cap X \neq \emptyset, u \in U\}, \quad (9)$$



$$[\underline{\mathbf{P}}(X)] = \{S_{P_i}(u) \mid S_{P_i}(u) \subseteq X_i, i \leq n, u \in U\}, \quad (10)$$

where  $X_1 = X$ ,  $X_i = X - \bigcup_{k=1}^{i-1} \underline{SIM}(P_k)(X_k)$  for  $i = 2, \dots, n$ , and  $[\cdot]$  denotes the structure of a rough approximation.

In the following, we show how the positive approximation in an incomplete information system works by an illustrate example.

Support  $S = (U, A)$  be an incomplete information system, where  $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$ ,  $P, Q \subseteq A$  two attribute sets,  $X = \{u_1, u_2, u_3, u_5, u_6\}$ ,  $SIM(P) = \{\{u_1, u_2\}, \{u_1, u_2\}, \{u_2, u_3\}, \{u_3, u_4, u_5\}, \{u_4, u_5, u_6\}, \{u_4, u_5, u_6\}\}$ ,  $SIM(Q) = \{\{u_1\}, \{u_2\}, \{u_3\}, \{u_4, u_5\}, \{u_4, u_5\}, \{u_5, u_6\}\}$ .

Obviously,  $P \succeq Q$  holds. Hence, we can construct a granulation order (a family of tolerance relations)  $\mathbf{P} = \{P, Q\}$ , where  $\mathbf{P}_1 = \{P\}$ ,  $\mathbf{P}_2 = \{P, Q\}$ .

By computing the positive approximation of  $X$  with respect to  $\mathbf{P}$ , we obtain easily that

$$\begin{aligned} [\underline{\mathbf{P}}_1(X)] &= \{\{u_1, u_2\}, \{u_1, u_2\}, \{u_2, u_3\}\} \\ [\overline{\mathbf{P}}_1(X)] &= \{\{u_1, u_2\}, \{u_1, u_2\}, \{u_2, u_3\}, \{u_3, u_4, u_5\}, \{u_4, u_5, u_6\}, \{u_4, u_5, u_6\}\}, \\ [\underline{\mathbf{P}}_2(X)] &= \{\{u_1, u_2\}, \{u_1, u_2\}, \{u_2, u_3\}, \{u_5, u_6\}\}, \\ [\overline{\mathbf{P}}_2(X)] &= \{\{u_1\}, \{u_2\}, \{u_3\}, \{u_4, u_5\}, \{u_4, u_5\}, \{u_5, u_6\}\}. \end{aligned}$$

Where  $\{u_1, u_2\}, \{u_2, u_3\}$  in  $[\underline{\mathbf{P}}_2(X)]$  are not induced by the tolerance relation  $SIM(Q)$  but  $SIM(P)$ , and  $[\overline{\mathbf{P}}_2(X)]$  is induced by the tolerance relation  $SIM(Q)$ . In other words, the target concept  $X$  is described by using the granulation order  $\mathbf{P} = \{P, Q\}$ .

In order to reveal the properties of positive approximation based on dynamic granulation in incomplete information systems, we introduce the notion of  $\sqsubseteq$ .

Assume  $A, B$  be two families of tolerance classes sets, where  $A = \{A_1, A_2, \dots, A_m\}$ ,  $B = \{B_1, B_2, \dots, B_n\}$ . We say  $A \sqsubseteq B$ , if and only if, for  $A_i \in A$ , there exists  $B_j \in B$  such that  $A_i \subseteq B_j$  ( $i \leq m, j \leq n$ ).

**Theorem 3.** . . .  $S = (U, A)$  . . . . .  $X \subseteq U$  . . . . .  $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$  . . . . .  $P_1 \succeq P_2 \succeq \dots \succeq P_n$  . . . . .  $P_i = \{P_1, P_2, \dots, P_i\}$  . . . . .  $[\underline{SIM}(P_i)(X)] \sqsubseteq [\underline{\mathbf{P}}_i(X)]$

**Remark.** Theorem 3 states that there is an inclusion relationship between the structure of the classical lower approximation  $\underline{SIM}(P_i)(X)$  and the structure of this new lower approximation  $\underline{\mathbf{P}}_i(X)$  based on a granulation order. In fact, for approximating a target concept, this mechanism establishes a family of tolerance classes with a hierarchy nature from rough to fine on the basis of keeping the approximation measure. Hence, in a board sense, the positive approximation will be helpful for extracting decision rules with hierarchy nature according to user requirements in incomplete information systems.

In the following, we introduce an approach to building a granulation order in an incomplete information system. As we know, the tolerance classes induced by an attribute set are finer than those of induced by any attribute subset in general. This idea can be used to build a granulation order from rough to fine on attribute power set. It can be understood by the below theorem.

**Theorem 4.** . . .  $S = (U, A)$  . . . . .  $A = \{a_1, a_2, \dots, a_n\}$  . . . . .  $A_i = \{a_1, a_2, \dots, a_i\} \quad i \leq n$  . . . . .  $\mathbf{P} = \{A_1, A_2, \dots, A_n\}$  . . . . .  $f_i$

In practical issues, a granulation order on attribute set can be appointed by user or experts, or be built according to the significance of each attribute. In particular, in an incomplete decision table (i.e., an incomplete information system with a decision attribute), some certain/uncertain decision rules can be extracted through constructing the positive approximation of a target decision.

Let  $S = (U, C \cup D)$  be an incomplete decision table,  $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$  a family of attribute sets with  $P_1 \succeq P_2 \succeq \dots \succeq P_n$ .  $\Gamma = U/D = \{D_1, D_2, \dots, D_r\}$  be a decision (partition) on  $U$ , a lower approximation and an upper approximation of  $\Gamma$  related to  $\mathbf{P}$  are defined by

$$\begin{aligned} \underline{\mathbf{P}}\Gamma &= \{\underline{\mathbf{P}}(D_1), \underline{\mathbf{P}}(D_2), \dots, \underline{\mathbf{P}}(D_r)\}, \\ \overline{\mathbf{P}}\Gamma &= \{\overline{\mathbf{P}}(D_1), \overline{\mathbf{P}}(D_2), \dots, \overline{\mathbf{P}}(D_r)\}. \end{aligned}$$

In addition, we call  $[bn_{\mathbf{P}}\Gamma] = \{\overline{\mathbf{P}}(D_i) - \underline{\mathbf{P}}(D_i) : i \leq r\}$   $\mathbf{P}$ -boundary region of  $\Gamma$ . Note that tolerance classes in  $\underline{\mathbf{P}}\Gamma$  can induce certain decision rules, while those in  $[bn_{\mathbf{P}}\Gamma]$  can extract uncertain decision rules from an incomplete decision table.

Similar to the formula (7), in the following, we give the notion of approximation measure of a target decision under a granulation order in an incomplete decision table.

**Definition 3.** . . .  $S = (U, C \cup D)$  . . . . .  $\Gamma = U/D = \{D_1, D_2, \dots, D_r\}$  . . . . .  $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$  . . . . .  $P_1 \succeq P_2 \succeq \dots \succeq P_n$  . . . . .  $P_i \in 2^C$  . . . . .  $\alpha_{\mathbf{P}}(\Gamma)$  . . . . .  $f_i$

$$\alpha_{\mathbf{P}}(\Gamma) = \sum_{k=1}^r \frac{|D_k| |\underline{\mathbf{P}}(D_k)|}{|U| |\overline{\mathbf{P}}(D_k)|}. \quad (11)$$

**Theorem 5.** . . .  $S = (U, C \cup D)$  . . . . .  $\Gamma = U/D = \{D_1, D_2, \dots, D_r\}$  . . . . .  $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$  . . . . .  $P_1 \succeq P_2 \succeq \dots \succeq P_n$  . . . . .  $P_i \in 2^C$  . . . . .  $\mathbf{P}_i = \{P_1, P_2, \dots, P_i\}$  . . . . .  $f_i$

$$\alpha_{\mathbf{P}_1}(\Gamma) \leq \alpha_{\mathbf{P}_2}(\Gamma) \leq \dots \leq \alpha_{\mathbf{P}_n}(\Gamma). \quad (12)$$

Theorem 5 states that the approximation measure  $\alpha_{\mathbf{P}}(\Gamma)$  increases as the granulation order become longer through adding attribute subsets.

### 3 Conclusions

In this paper, we have extended rough set approximation under static granulation to rough set approximation under dynamic granulation in the context of incomplete information systems, and its some properties have been investigated. A target concept can be approached by the change of the positive approximation. The results obtained in this paper will be helpful for further research on rough set theory and its practical applications.

**Acknowledgements.** This work was supported by the national high technology research and development program (No. 2007AA01Z165), the national natural science foundation of China (No. 70471003, No. 60573074), the foundation of doctoral program research of the ministry of education of China (No. 20050108004), key project of science and technology research of the ministry of education of China (No. 206017) and the graduate student innovation foundation of Shanxi.

## References

1. Zadeh, L.A.: Fuzzy Sets and Information Granularity. In: Gupta, M., Ragade, R., Yager, R. (eds.) *Advances in Fuzzy Set Theory and Application*, pp. 3–18. North-Holland, Amsterdam (1979)
2. Zadeh, L.A.: Fuzzy logic=computing with words. *IEEE Transactions on Fuzzy Systems* 4(1), 103–111 (1996)
3. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems* 90, 111–127 (1997)
4. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information / intelligent systems. *Soft Computing* 2(1), 23–25 (1998)
5. Pawlak, Z.: Granularity of knowledge, indiscernibility and rough sets. In: *Proceedings of 1998 IEEE International Conference on Fuzzy Systems*, pp. 106–110. IEEE Computer Society Press, Los Alamitos (1998)
6. Yao, Y.Y.: Granular computing: basic issues and possible solutions. In: *Proceedings of the Fifth International Conference on Computing and Information*, vol. I, pp. 186–189 (2000)
7. Lin, T.Y.: Granular computing on binary relations I: Data mining and neighborhood systems, II: Rough sets representations and belief functions. In: Polkowski, L., Skowron, A. (eds.) *Rough Sets in Knowledge Discovery 1*, pp. 107–140. Physica, Heidelberg (1998)
8. Zhang, L., Zhang, B.: Theory of fuzzy quotient space (methods of fuzzy granular computing). *Journal of Software* (in Chinese) 14(4), 770–776 (2003)
9. Klir, G.J.: Basic issues of computing with granular computing. In: *Proceedings of 1998 IEEE International Conference on Fuzzy Systems*, pp. 101–105. IEEE Computer Society Press, Los Alamitos (1998)
10. Liang, J.Y., Shi, Z.Z.: The information entropy, rough entropy and knowledge granulation in rough set theory. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 12(1), 37–46 (2004)
11. Liang, J.Y., Shi, Z.Z., Li, D.Y.: The information entropy, rough entropy and knowledge granulation in incomplete information systems. *International Journal of General Systems* 35(6), 641–654 (2006)
12. Liu, Q.: Granules and applications of granular computing in logical reasoning. *Journal of Computer Research and Development* (in Chinese) 41(4), 546–551 (2004)
13. Liang, J.Y., Qian, Y.H., Chu, C.Y., Li, D.Y., Wang, J.H.: Rough set approximation based on dynamic granulation. In: Ślezak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) *RSFDGrC 2005. LNCS (LNAI)*, vol. 3641, pp. 701–708. Springer, Heidelberg (2005)
14. Krysziakiewicz, M.: Rough set approach to incomplete information system. *Information Sciences* 112, 39–49 (1998)

15. Kryszkiewicz, M.: Rule in incomplete information systems. *Information Sciences* 113, 271–292 (1999)
16. Kryszkiewicz, M.: Comparative study of alternative type of knowledge reduction in inconsistent systems. *International Journal of Intelligent systems* 16, 105–120 (2001)
17. Leung, Y., Li, D.Y.: Maximal consistent block technique for rule acquisition in incomplete information systems. *Information Sciences* 153, 85–106 (2003)
18. Pawlak, Z.: *Rough sets. Theoretical aspects of reasoning about data.* Kluwer Academic Publishers, Dordrecht (1991)
19. Liang, J.Y., Li, D.Y.: *Uncertainty and knowledge acquisition in information systems (in Chinese).* Science Press, Beijing (2005)

# Generalized Fuzzy Operations for Digital Hardware Implementation

Ildar Batyrshin<sup>1</sup>, Antonio Hernández Zavala<sup>2</sup>, Oscar Camacho Nieto<sup>2</sup>,  
and Luis Villa Vargas<sup>2</sup>

<sup>1</sup> Mexican Petroleum Institute  
batyr1@gmail.com

<sup>2</sup> Mexican Polytechnic Institute – Computer Research Centre  
antonioh@hotmail.com, oscarcc@cic.ipn.mx, lvilla@cic.ipn.mx

**Abstract.** Hardware implementation of fuzzy systems plays important role in many industrial applications of fuzzy logic. The most popular applications of fuzzy hardware systems were found in the domain of control systems but the area of application of these systems is extending on other areas such as signal processing, pattern recognition, expert systems etc. The digital fuzzy hardware systems usually use only basic operations of fuzzy logic like min, max and some others, first, due to their popularity in traditional fuzzy control systems and, second, due to the difficulties of hardware implementation of more complicated operations, e.g. parametric classes of  $t$ -norms and  $t$ -conorms. But for extending the area of applications and flexibility of fuzzy hardware systems it is necessary to develop the methods of digital hardware implementation of wide range of fuzzy operations. The paper studies the problem of digital hardware implementation of fuzzy parametric conjunction and disjunction operations. A new class of such operations is proposed which is simple for digital hardware implementation and is flexible, due to its parametric form, for possible tuning in fuzzy models. The methods of hardware implementation of these operations in digital systems are proposed.

**Keywords:** Fuzzy system, conjunction, disjunction, digital system.

## 1 Introduction

Fuzzy logic gives possibility to translate human knowledge into rules of fuzzy systems. Such systems have wide applications often providing better results than conventional techniques [1-3]. For on-board and real-time applications the fuzzy systems require faster processing speed which makes specific inference hardware the choice to satisfy processing time demands. Hardware implementation of fuzzy systems plays an important role in many industrial applications of fuzzy logic, mainly in the domain of control systems, but the area of application of these systems is extending on other areas such as signal processing, pattern recognition, expert systems etc [4 -6]. Departure point for digital fuzzy processors was at mid 80's by Togai and Watanabe [7] and, since then, many architectures have been presented with faster inferences and decreasing hardware complexity [6, 8-11]. FPGA technology

has become a fast design alternative to implement complex digital systems, given its reprogrammable capabilities. This technology have been used by some researchers to adapt the mathematics of fuzzy logic systems into digital circuitry [12-15].

The digital fuzzy hardware systems usually use as fuzzy conjunction and disjunction only basic operations of fuzzy logic like *min*, *max* and product [16], first, due to their popularity in traditional fuzzy control systems and, second, due to the difficulties of hardware implementation of more complicated operations, e.g. parametric classes of *t*-norms and *t*-conorms [17]. Among mentioned operations, *min* and *max* are simple comparisons between two values, which on digital circuitry is easy to implement; product is much more complex, because of the following reasons [18,19]:

1. It requires at least  $n-1$  iterations on a sequential multiplier with  $n$  bits.
2. For the case of a combinatorial array circuit  $n-1$  levels of adders are required for each pair of  $n$  bits used for input length.
3. Shifting operation can be realized minimizing time and resources but only for multiplying a number by a power of two.

Fuzzy rule based systems are usually constructed based on expert knowledge and on experimental data describing system behavior. First fuzzy systems usually have been based on expert knowledge and trial-and-error adjustment of rules and membership functions. At the beginning of 90's it was proved that fuzzy systems are universal approximators [20,21]. This fundamental result in fuzzy set theory stimulated development of different methods of fuzzy systems optimization based on automatic adjustment of rules and membership functions [3, 22, 23]. Another approach to fuzzy system optimization was proposed in [24, 25] where instead of adjusting of membership functions it was proposed to adjust parameters of fuzzy operations used in fuzzy systems. This approach gives possibility to keep unchanged expert knowledge about fuzzy concepts given in membership functions. Instead of traditional *t*-norms and *t*-conorms sufficiently complicated for tuning in optimization process it was proposed to introduce simple parametric fuzzy conjunction and disjunction operations satisfying simplified system of axioms [24,25] in contrast to *t*-norms and *t*-conorms satisfying very restrictive associativity property [17].

In this paper it is considered a problem of hardware implementation of fuzzy systems obtained as a result of adjusting of parametric conjunction and disjunction operations. It should be noted that both parametric *t*-norms [17] and parametric generalized conjunctions considered in [24, 25] use the product operation as a constituent. As it was mentioned above the product operation has not sufficiently efficient hardware implementation. To avoid this problem this paper introduces new parametric family of fuzzy conjunction operations without product operation as a constituent. Disjunction operations can be obtained dually to conjunctions operations.

The paper has the following structure. Section 2 gives the basic definitions of fuzzy conjunction and disjunction operations. In Section 3 a new method of generation of non-associative conjunctions suitable for hardware implementation is proposed. The methods of a hardware implementation of proposed parametric conjunction operations are considered in Section 4. In Conclusion we discuss obtained results and future directions of research.

## 2 Basic Definitions

Triangular norm ( $t$ -norm)  $T$  and triangular conorm ( $t$ -conorm)  $S$  are defined as functions  $T, S: [0,1] \times [0,1] \rightarrow [0,1]$  satisfying on  $[0,1]$  the following axioms [17]:

$$T(x,y) = T(y,x), \quad S(x,y) = S(y,x), \quad (\text{commutativity})$$

$$T(T(x,y),z) = T(x,T(y,z)), \quad S(S(x,y),z) = S(x,S(y,z)), \quad (\text{associativity})$$

$$T(x,y) \leq T(u,v), \quad S(x,y) \leq S(u,v), \quad \text{if } x \leq u, y \leq v \quad (\text{monotonicity})$$

$$T(x,1) = x, \quad S(x,0) = x \quad (\text{boundary conditions})$$

From this definition the following properties are followed:

$$T(0,x) = T(x,0) = 0, \quad S(1,x) = S(x,1) = 1, \quad T(1,x) = T(x,1) = x, \quad S(0,x) = S(x,0) = x \quad (1)$$

$t$ -norm and  $t$ -conorm are dual.

An involutive negation is a function  $N: [0,1] \rightarrow [0,1]$  satisfying on  $[0,1]$  the following conditions:

$$n(x) \leq n(y) \quad \text{if } y \leq x,$$

$$n(0) = 1, \quad n(1) = 0,$$

$$n(n(x)) = x.$$

$t$ -norm and  $t$ -conorm can be obtained one from another by means of negation operation as follows:

$$S(x,y) = n(T(n(x),n(y))), \quad T(x,y) = n(S(n(x),n(y))).$$

The following are the simplest  $t$ -norm and  $t$ -conorm mutually related by De Morgan laws with negation operation  $n(x) = 1 - x$ :

$$T_M(x,y) = \min\{x,y\} \quad (\text{minimum}), \quad S_M(x,y) = \max\{x,y\}, \quad (\text{maximum}),$$

$$T_P(x,y) = x \cdot y \quad (\text{product}), \quad S_P(x,y) = x + y - x \cdot y, \quad (\text{probabilistic sum}),$$

$$T_L(x,y) = \max\{x+y-1, 0\}, \quad S_L(x,y) = \min\{x+y, 1\}, \quad (\text{Lukasiewicz}),$$

$$T_D(x,y) = \begin{cases} 0, & \text{if } (x,y) \in [0,1] \times [0,1] \\ \min(x,y), & \text{otherwise} \end{cases}, \quad (\text{drastic product}),$$

$$S_D(x,y) = \begin{cases} 1, & \text{if } (x,y) \in (0,1] \times (0,1] \\ \max(x,y), & \text{otherwise} \end{cases} \quad (\text{drastic sum})$$

Lukasiewicz  $t$ -norm  $T_L$  and  $t$ -conorm  $S_L$  are also called a bounded product and bounded sum, respectively.

All  $t$ -norms  $T$  and  $t$ -conorms  $S$  satisfy on  $[0,1]$  the following inequalities:

$$T_D(x,y) \leq T(x,y) \leq T_M(x,y) \leq S_M(x,y) \leq S(x,y) \leq S_D(x,y). \tag{2}$$

As follows from these inequalities  $T_D(x,y)$ ,  $T_M(x,y)$ ,  $S_M(x,y)$  and  $S_D(x,y)$  serve as boundaries for all  $t$ -norms and  $t$ -conorms.

Several families of parametric  $t$ -norms and  $t$ -conorms can be found in [17]. Below is an example of Dombi  $t$ -norm, depending on parameter  $\lambda \in [0, \infty]$ :

$$T(x,y) = \frac{1}{1 + \left( \left( \frac{1-x}{x} \right)^\lambda + \left( \frac{1-y}{y} \right)^\lambda \right)^{\frac{1}{\lambda}}} \quad \text{if } \lambda \in (0, \infty),$$

$$T(x,y) = T_D(x,y), \quad \text{if } \lambda = 0,$$

$$T(x,y) = T_M(x,y), \quad \text{if } \lambda = \infty.$$

As it was noted in [24], parametric  $t$ -norms and  $t$ -conorms have sufficiently complicated form due to the associativity property requiring to use inverse functions of generators of these operations [17]. For this reason, traditional parametric  $t$ -norms are sufficiently complicated for automatic adjusting of their parameters in automatic optimization of fuzzy systems. To obtain more simple parametric  $t$ -norms in [24,25] it was proposed to use non-associative conjunction operations. Usually the property of associativity does not used in construction of applied fuzzy systems where position of operands of these operations is fixed. Moreover, often only two operands are used in these operations as in fuzzy control systems with two input variables. Several methods of parametric non-associative conjunctions were proposed in [24]. One of such methods is following:

$$T(x,y) = T_2(T_1(x,y), S(g_1(x), g_2(y)))$$

where  $T_1$  and  $T_2$  are some conjunctions,  $S$  is a disjunction, and  $g_1, g_2$  are non-decreasing functions  $g_1, g_2: [0,1] \rightarrow [0,1]$  such that  $g_1(1) = g_2(1) = 1$ . As  $T_1$  and  $T_2$  it can be used for example one of basic  $t$ -norms considered above. Some examples of simple parametric conjunctions obtained by this method are following:

$$T(x,y) = \begin{cases} \min(x,y), & \text{if } p \leq x \text{ or } q \leq y \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

$$T(x,y) = \min(x,y) \cdot \max\{1 - p(1-x), 1 - q(1-y), 0\}, \tag{4}$$

$$T(x,y) = \min\{\min(x,y), \max(x^p, y^q)\}, \tag{5}$$

$$T(x,y) = \min(x,y) \cdot \max(x^p, y^q).$$



Note that inequality (2) is also fulfilled for non-associative conjunctions.

In [25] more generalized conjunction operations defined by monotonicity property and simplified boundary conditions:

$$T(0,0) = T(0,1) = T(1,0) = 0, \quad T(1,1) = 1,$$

were considered. The following methods of generation of such operations were proposed:

$$T(x,y) = T_2(T_1(x,y), S_1(g_1(x), g_2(y))),$$

$$T(x,y) = T_2(T_1(x,y), g_1(S_1(x,y))),$$

$$T(x,y) = T_2(T_1(x,y), S_2(h(x), S_1(x,y))),$$

where  $S$  is a monotone function satisfying conditions:

$$S(1,0) = S(0,1) = S(1,1) = 1$$

and  $g_1, g_2, h$  are non-decreasing functions  $g_1, g_2, h: [0,1] \rightarrow [0,1]$  such that  $g_1(1) = g_2(1) = h(1) = 1$ . These methods gives possibility to generate the following simplest conjunction operations :

$$T(x,y) = \min(x^p, y^q),$$

$$T(x,y) = x^p y^q,$$

$$T(x,y) = (xy)^p (x + y - xy)^q.$$

Parametric conjunctions introduced in [24,25] are simpler than most of known parametric  $t$ -norms and suitable for their adjusting in optimization of fuzzy models but hardware implementation of most of these operations is still non-effective due to the presence of operations product and computing powers in their definitions. For this reason only parametric operation (3) considered above can have effective hardware implementation. In the following section we propose a new method of generation of non-associative conjunctions which can give possibility to construct parametric conjunctions based only on basic  $\min$ ,  $\max$ , Lukasiewicz and drastic operations which have effective hardware implementation.

### 3 New Method of Generation of Non-associative Conjunctions

We propose the following new method of generation of conjunctions:

$$T(x,y) = \min(T_1(x,y), S(T_2(x,y), s)), \quad (6)$$

where  $T_1$  and  $T_2$  are some conjunctions,  $S$  is a disjunction, and  $s$  is a parameter  $s \in [0,1]$ .

The following properties of (6) can be proved.

**Theorem 1.** If  $T_1$ ,  $T_2$  and  $S$  are commutative, monotonic functions satisfying boundary conditions (1) then  $T$  is the same.

**Proposition 2.** For specific  $t$ -norms and  $t$ -conorms (6) is reduced as follows:

$$\text{if } T_2 = T_M \text{ then } T(x,y) = T_1(x,y);$$

$$\text{if } T_1 = T_D \text{ then } T(x,y) = T_D(x,y);$$

$$\text{if } T_1 = T_2 \text{ then } T(x,y) = T_1(x,y).$$

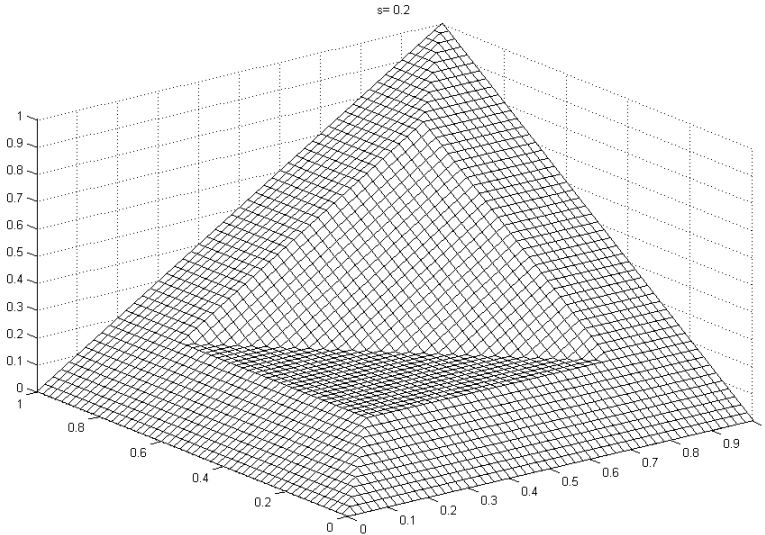
Based on (6) and avoiding cases considered in Proposition 2 by means of basic  $t$ -norms and  $t$ -conorms  $T_M$ ,  $T_L$  and  $T_D$  we can obtain new parametric conjunctions with effective hardware implementation. For example we can introduce the following simple parametric conjunctions:

$$T(x,y) = \min(\min(x,y), S_L(T_L(x,y),s)), \tag{7}$$

$$T(x,y) = \min(\min(x,y), S_L(T_{MB}(x,y),s)), \tag{8}$$

where  $T_{MB}$  denotes a conjunction (3). Fig. 1 depicts the shape of conjunction (7) and Fig. 2 depicts the shape of conjunction (8). In these pictures parameters  $s, p, q$  define the sizes of the “holes” in the “pyramid” corresponding to the conjunction  $T_M$ .

In the following section we consider examples of hardware implementation of some new parametric conjunctions.



**Fig. 1.** The shape of the conjunction (7)

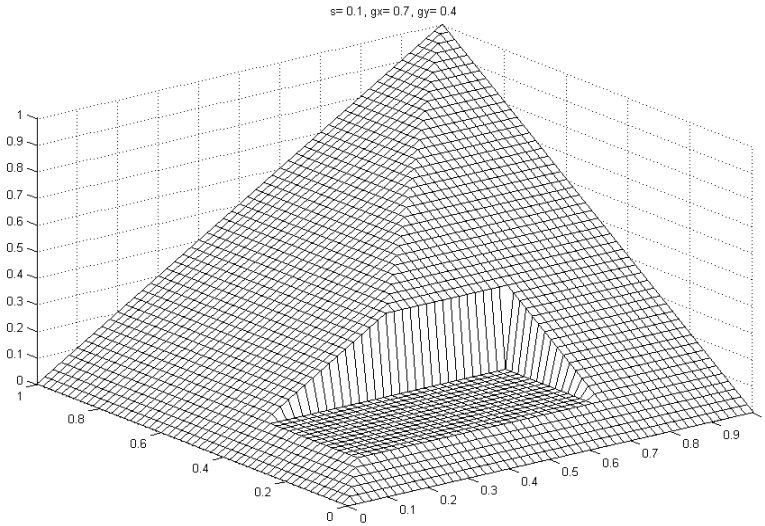


Fig. 2. The shape of the conjunction (8)

## 4 Hardware Implementation of Fuzzy Conjunctions

Fuzzy membership value is commonly expressed as a floating point number in interval  $[0, 1]$ , where 0 means non membership and 1 means complete membership and generally can have infinite values. Unfortunately this is not so easy for a computer to do calculations by using this representation, instead of this we can work with integers between  $[0, 2^n-1]$ , where  $n$  is the number of bits used to represent truth space and gives the resolution, 0 means no membership, and  $2^n-1$  is the complete membership value. We consider here 8 bit resolution of numbers. Denote  $d = 2^n-1$ ,  $n=8$ .

Below are the methods of hardware implementation of Lukasiewicz t-norm and t-conorm (bounded product and bounded sum) are discussed. These circuits were realized using Xilinx tools for FPGA design, obtaining equivalent circuits for mentioned operations. Basic logical gates [26] are used to construct the circuits, some operations used are common constructs on digital hardware this is the case of comparator, adder and subtractor [18,19]. The hardware implementation of bounded product is shown in Fig. 3.

In the first block from left, there is an adder to implement  $x+y$ , if the sum of this values is more than 8 bit value count there is a carry output flag to indicate that data is not valid. Second block subtract  $d$  from result of previous block, there is also a carry output flag to identify when data is not valid. In order to obtain a valid data up to this part it is necessary to have two valid conditions on two previous blocks, this correspond to  $CO=0$  on the addition block and  $CO=1$  on the subtraction block, this is realized by an AND gate. Third and fourth blocks correspond to the maximum operation between 0 and valid result from previous stage, AND gate between both blocks is on charge to decide if there is no valid data, let 0 be the output value.

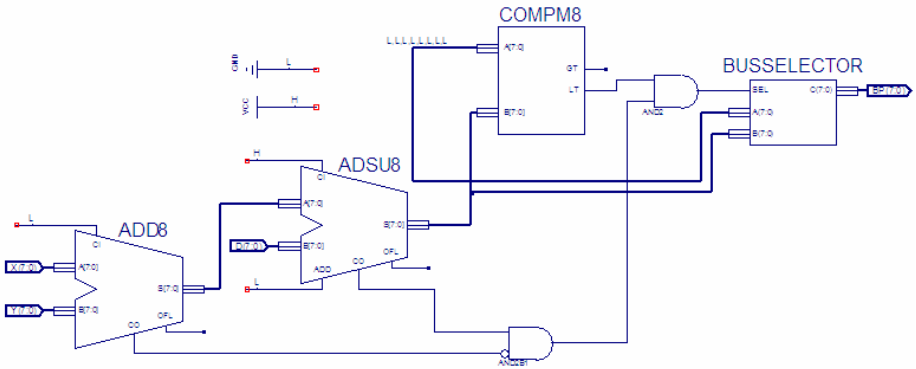


Fig. 3. Digital hardware for bounded product

Circuit shown in Fig.4 corresponds to a bounded sum operation. Its functioning is alike previous circuit in two first blocks, the main difference here is that last two blocks perform minimum operation.

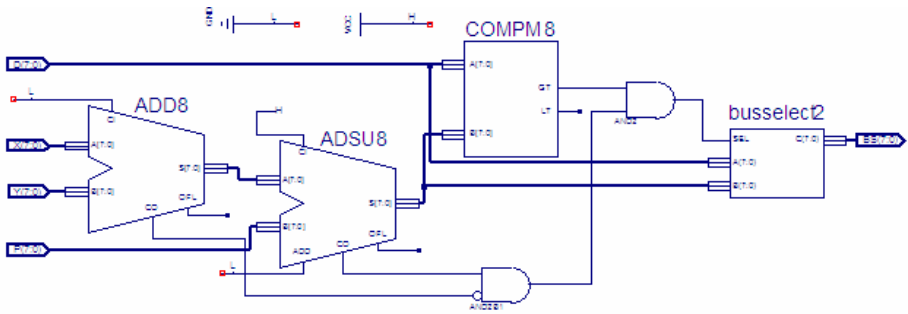


Fig. 4. Digital hardware for bounded sum operation

A new parametric conjunction (7) is represented in Fig. 5. Here, each block corresponds to the circuit diagrams shown before.

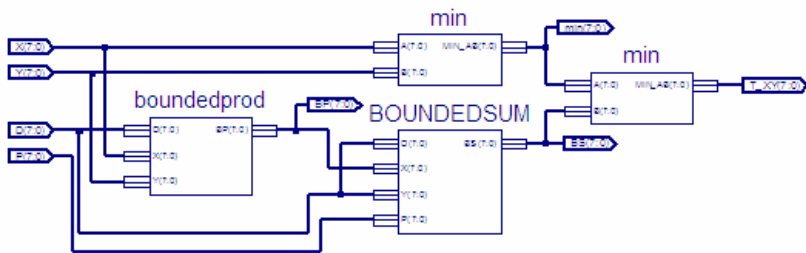


Fig. 5. Hardware implementation of parametric conjunction (7)

## 5 Conclusions

The main contribution of the paper is the following. The problem of effective digital hardware implementation of fuzzy parametric conjunction and disjunction operations is formulated and its solution is proposed. It is a first step in hardware implementation of fuzzy systems obtained as a result of adjusting of parameters of operations. Most of known parametric fuzzy conjunction and disjunction operations have not effective hardware implementation because they use product and computing powers operations in their definitions. New method of generation of parametric conjunction and disjunction operations is proposed in this paper. Based on this method new parametric classes of fuzzy operations, suitable for effective hardware implementation, are obtained. The methods of hardware implementation of some of these operations are given. The obtained results can be extended in several directions. First, effective hardware implementation of parametric operations based on drastic  $t$ -norm and  $t$ -corm can be done in the similar manner as it was done for operations based on Lukasiewicz operations. Second, obtained results can be used in digital hardware implementation of inference and aggregation operations in fuzzy systems with parametric conjunctions and disjunctions. Hardware implementation of such systems will extend possibilities of design of flexible on-board and real-time fuzzy systems.

**Acknowledgments.** The research work was partially supported by CONACYT and SIP – IPN, Project No. 20070945.

## References

1. Terano, T., Asai, K., Sugeno, M.: Applied Fuzzy Systems. Academic Press Professional, San Diego (1994)
2. Yen, J., Langari, R., Zadeh, L.A.: Industrial Applications of Fuzzy Logic and Intelligent Systems. IEEE Press, NJ (1995)
3. Jang, J.-S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence (1997)
4. Jespers, P.G.A., Dualibe, C., Verleysen, M.: Design of Analog Fuzzy Logic Controllers in CMOS Technologies. In: Implementation, Test and Application, Kluwer Academic Publishers, New York (2003)
5. Kandel, A., Langholz, G.: Fuzzy Hardware: Architectures and Applications. Kluwer Academic Publishers, Dordrecht (1997)
6. Patyra, M.J., Grantner, J.L., Koster, K.: Digital Fuzzy Logic Controller: Design and Implementation. IEEE Transactions on Fuzzy Systems 4, 439–459 (1996)
7. Togai, M., Watanabe, H.: A VLSI Implementation of a Fuzzy-Inference Engine: Toward an Expert System on a Chip. Information Sci. 38, 147–163 (1986)
8. Cardarilli, G.C., Re, M., Lojacono, R., Salmeri, M.: A New Architecture for High-Speed COG Based Defuzzification. In: TOOLMET 1997. International Workshop on Tool Environments and Development Methods for Intelligent Systems, pp. 165–172 (1997)
9. Gaona, A., Olea, D., Melgarejo, M.: Distributed Arithmetic in the Design of High Speed Hardware Fuzzy Inference Systems. In: 22nd International Conference of the North American Fuzzy Information Processing Society, pp. 116–120 (2003)

10. Banaiyan, A., Fakhraie, S.M., Mahdiani, H.R.: Cost-Performance Co-Analysis in VLSI Implementation of Existing and New Defuzzification Methods. In: Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, vol. 1, pp. 828–833 (2005)
11. Tamukoh, H., Horio, K., Yamakawa, T.: A Bit-Shifting-Based Fuzzy Inference for Self-Organizing Relationship (SOR) Network. *IEICE Electronics Express* 4, 60–65 (2007)
12. Salapura, V., Hamann, V.: Implementing Fuzzy Control Systems Using VHDL and Statecharts. In: EURO-DAC 1996 with EURO-VHDL 1996. Proc. of the European Design Automation Conference, pp. 53–58. IEEE Computer Society Press, Geneva (1996)
13. Sánchez-Solano, S., Senhadji, R., Cabrera, A., Baturone, I., Jiménez, C.J., Barriga, A.: Prototyping of Fuzzy Logic-Based Controllers Using Standard FPGA Development Boards. In: Proc. IEEE International Workshop on Rapid System Prototyping, pp. 25–32. IEEE Computer Society Press, Los Alamitos (2002)
14. Garrigos-Guerrero, F.J., Ruiz Merino, R.: Implementacion de Sistemas Fuzzy Complejos sobre FPGAs. In: Computación Reconfigurable y FPGAs, pp. 351–358 (2003)
15. Raychev, R., Mtibaa, A., Abid, M.: VHDL Modelling of a Fuzzy Co-Processor Architecture. In: International Conference on Computer Systems and Technologies – CompSysTech (2005)
16. Zadeh, L.A.: Fuzzy Sets. *Information and Control* 8, 338–353 (1965)
17. Klement, E.P., Mesiar, R., Pap, E.: *Triangular Norms*. Kluwer, Dordrecht (2000)
18. Patterson, D.A., Hennessy, J.L.: *Computer Organization and Design: The Hardware/Software Interface*, 2nd edn. Morgan Kaufmann Publishers, San Francisco (1998)
19. Zargham, M.R.: *Computer Architecture: Single and Parallel Systems*. Prentice-Hall, Englewood Cliffs (1995)
20. Kosko, B.: Fuzzy Systems as Universal Approximators. In: Proc. IEEE Int. Conf. on Fuzzy Systems, pp. 1153–1162. IEEE Computer Society Press, Los Alamitos (1992)
21. Wang, L.X.: Fuzzy Systems are Universal Approximators. In: Proc. IEEE Int. Conf. On Fuzzy Systems, pp. 1163–1170. IEEE Computer Society Press, Los Alamitos (1992)
22. Kosko, B.: *Fuzzy Engineering*. Prentice-Hall, New Jersey (1997)
23. Wang, L.-X.: *A Course in Fuzzy Systems and Control*. Prentice Hall PTR, Upper Saddle River, NJ (1997)
24. Batyrshin, I., Kaynak, O.: Parametric Classes of Generalized Conjunction and Disjunction Operations for Fuzzy Modeling. *IEEE Transactions on Fuzzy Systems* 7, 586–596 (1999)
25. Batyrshin, I., Kaynak, O., Rudas, I.: Fuzzy Modeling Based on Generalized Conjunction Operations. *IEEE Transactions on Fuzzy Systems* 10, 678–683 (2002)
26. Tocci, R.J., Widmer, N.S., Moss, G.L.: *Digital Systems: Principles and Applications*, 9th edn. Prentice-Hall, Englewood Cliffs (2003)

# A Novel Model of Artificial Immune System for Solving Constrained Optimization Problems with Dynamic Tolerance Factor

Victoria S. Aragón, Susana C. Esquivel<sup>1</sup>, and Carlos A. Coello Coello<sup>2</sup>

<sup>1</sup> Laboratorio de Investigación y Desarrollo en Inteligencia Computacional\*

Universidad Nacional de San Luis

Ejército de los Andes 950

(5700) San Luis, Argentina

{vsaragon, esquivel}@unsl.edu.ar

<sup>2</sup> CINVESTAV-IPN (Evolutionary Computation Group)\*\*

Departamento de Computación

Av. IPN No. 2508, Col. San Pedro Zacatenco

México D.F. 07300, México

ccoello@cs.cinvestav.mx

**Abstract.** In this paper, we present a novel model of an artificial immune system (AIS), based on the process that suffers the T-Cell. The proposed model is used for solving constrained (numerical) optimization problems. The model operates on three populations: Virgins, Effectors and Memory. Each of them has a different role. Also, the model dynamically adapts the tolerance factor in order to improve the exploration capabilities of the algorithm. We also develop a new mutation operator which incorporates knowledge of the problem. We validate our proposed approach with a set of test functions taken from the specialized literature and we compare our results with respect to Stochastic Ranking (which is an approach representative of the state-of-the-art in the area) and with respect to an AIS previously proposed.

## 1 Introduction

In many real-world problems, the decision variables are subject to a set of constraints, and the search has to be bounded accordingly. Constrained optimization problems are very common, for example, in engineering applications, and therefore the importance of being able to deal with them efficiently.

Many bio-inspired algorithms (particularly evolutionary algorithms) have been very successful in the solution of a wide variety of optimization problems [1]. But, when they are used to solve constrained optimization problems, they need a special method to incorporate the problem's constraints into their fitness

---

\* LIDIC is financed by Universidad Nacional de San Luis and ANPCyT (Agencia Nacional para promover la Ciencia y Tecnología).

\*\* The third author acknowledges support from CONACyT project no. 45683-Y.

function. Evolutionary algorithms (EAs) often use exterior penalty functions in order to do this [2]. However, penalty functions require the definition of accurate penalty factors and performance is highly dependent on them.

The main motivation of the work presented in this paper is to explore the capabilities of a new AIS model in the context of constrained global optimization. The proposed model is based on the process that suffers the T-Cell. We also propose a dynamic tolerance factor and several mutation operators that allow us to deal with different types of constraints.

## 2 Statement of the Problem

We are interested in solving the general nonlinear programming problem which is defined as follows:

Find  $\mathbf{x} = (x_1, \dots, x_n)$  which optimizes  $f(x_1, \dots, x_n)$  subject to:

$$\begin{aligned} h_i(x_1, \dots, x_n) &= 0 \quad i = 1, \dots, l \\ g_j(x_1, \dots, x_n) &\leq 0 \quad j = 1, \dots, p \end{aligned}$$

where  $(x_1, \dots, x_n)$  is the vector of solutions (or decision variables),  $l$  is the number of equality constraints and  $p$  is the number of inequality constraints (in both cases, constraints could be linear or nonlinear).

## 3 Previous Work

According to [3] the main models of Artificial Immune System are: Negative Selection, Clonal Selection and Immune Network Models. They are briefly described next.

Forrest et al. [4] proposed the Negative Selection model for detection of changes. This model is based on the discrimination principle that the immune system adopts to distinguish between self and nonself. This model generates random detectors and discards the detectors that are unable of recognizing themselves. Thus, it maintains the detectors that identify any nonself. It performs a probabilistic detection and it is robust because it searches any foreign action instead of a particular action.

The Immune Network Model was proposed by Jerne [5], and it is a mathematical model of the immune system. In this case, the dynamics of the lymphocytes are simulated by differential equations. This model assumes that lymphocytes are an interconnected network. Several models have been derived from it [6,7].

Clonal Selection is based on the way in which both B-cells and T-cells adapt in order to match and kill the foreign cells [3]. Clonal Selection involves: 1) the AIS' ability to adapt its B-cells to new types of antigens and 2) the affinity maturation by hypermutation. CLONALG proposed by Nunes de Castro and Von Zuben [8] was originally used to solve pattern recognition and multimodal optimization problems, and there are a few extensions of this algorithm for constrained optimization. CLONALG works in the following way: first, it creates a



random population of antibodies, it sorts it according to some fitness function, it clones them, it mutates each clone, it selects the fittest antibody and clones it and replaces the worst antibodies for antibodies that are randomly generated.

Those models have been used in several types of problems, but particularly, the use of artificial immune systems to solve constrained (numerical) optimization problems is scarce. The only previous related work that we found in the specialized literature is the following:

Hajela and Yoo [1] have proposed a hybrid between a Genetic Algorithm (GA) and an AIS for solving constrained optimization problems. This approach works on two populations. The first is composed by the antigens (which are the best solutions), and the other by the antibodies (which are the worst solutions). The idea is to have a GA embedded into another GA. The outer GA performs the optimization of the original (constrained) problem. The second GA uses as its fitness function a Hamming distance so that the antibodies are evolved to become very similar to the antigens, without becoming identical. An interesting aspect of this work was that the infeasible individuals would normally become feasible as a consequence of the evolutionary process performed.

Kelsey and Timmis [9] proposed an immune inspired algorithm based on the clonal selection theory to solve multimodal optimization problems. Its highlight is the mutation operator called *Somatic Contiguous Hypermutation*, where mutation is applied on a subset of contiguous bits. The length and beginning of this subset is determined randomly.

Coello Coello and Cruz-Cortés [10] have proposed an extension of Hajela and Yoo's algorithm. In this proposal, no penalty function is needed, and some extra mechanisms are defined to allow the approach to work in cases in which there are no feasible solutions in the initial population.

Luh and Chueh [11] have proposed an algorithm (called CMOIA, or Constrained Multi Objective Immune Algorithm) for solving constrained multiobjective optimization problems. In this case, the antibodies are the potential solutions to the problem, whereas antigens are the objective functions. CMOIA transforms the constrained problem into an unconstrained one by associating an interleukine (IL) value with all the constraints violated. IL is a function of both the number of constraints violated and the total magnitude of this constraint violation. Then, feasible individuals are rewarded and infeasible individuals are penalized.

Coello Coello and Cruz-Cortés [12] have proposed an algorithm based on the clonal selection theory for solving constrained optimization problems. The authors experimented with both binary and real-value representation, considering Gaussian-distributed and Cauchy-distributed mutations. Furthermore, they proposed a controlled and uniform mutation operator.

## 4 Our Proposed Model

This paper presents a novel bio-inspired model based on the T-Cells; appropriately, it is called "T-Cell Model". In a very simple way, the processes that

suffer the T-Cells are the following: first, they are divided in three groups (Virgin Cells, Effector Cells and Memory Cells). Then, the natural immune system generates a huge number of virgin cells. During the immunological response, the T-cells pass through different phases: initiation, reaction and elimination. After the initiation phase, the virgin cells become effector cells. These react (the cells change in order to improve) and undergo a process called *apoptosis*. This process eliminates any undesirable cells. The surviving cells become memory cells.

Thus, this model operates on three populations, corresponding to the three groups in which the T-cells are divided: (1) Virgin Cells (VC), (2) Effector Cells (EC) and (3) Memory Cells (MC). Each of them has a specific function. VC has as its main goal to provide diversity. EC tries to explore the conflicting zones of the search space. MC has to explore the neighborhood of the best solutions found so far. The *apoptosis* is modeled through the insertion of VC's cells into EC and EC's cells into MC. VC and EC represent their cells with binary strings using Gray coding and MC does the same, but adopting vectors of real numbers. The general structure of this model is the following:

Repeat a predetermined number of times

1. Generate (in a random way) Virgin Cells
2. Insert a percentage of Virgin Cells in Effector Cells
3. Repeat a predetermined number of times
  - 3.1. Make the Effector Cells React

End repeat

4. Insert a percentage of Effectors Cells in Memory Cells
5. Repeat a predetermined number of times
  - 5.1. Make the Memory Cells React

End repeat

End repeat

#### 4.1 Handling Constraints

In our proposed model, the constraint-handling method needs to calculate, for each cell (solution), regardless of the population to which it belongs, the following: 1) the value of each constraint function, 2) the sum of constraints violation (sum\_res)<sup>1</sup> and 3) the value of the objective function (only if the cell is feasible).

When the search process is driven by the value of each constraint function and the sum of constraints violation, then the selection mechanism favors feasible solutions over the infeasible ones. In this case, it is probable that, in some functions, the search falls into a local optimum. For this reason, we developed a dynamic tolerance factor (DTF), which changes with each new population, since it depends on the value of sum\_res specific of the cells of the population considered (VC or EC). The DTF is calculated by adding the value of each constraint violated in each cell from a particular population (VC or EC). Then, this

---

<sup>1</sup> This is a positive value determined by  $g_i(x)^+$  for  $i = 1, \dots, p$  and  $|h_k(x)|$  for  $k = 1, \dots, l$ .

value is divided by the number of Virgin Cells (for DTF's VC) or three times the number of Effector Cells (for DTF's EC).

When we evaluate the population using the DTF, it will be easier to generate solutions that are considered "feasible" (although they may be really infeasible if evaluated with the actual precision required). DTF relaxes the tolerance factor in order to adapt it to the particular cell into a population. This allows the exploration of each solution's neighborhood, which otherwise, would not be possible. This DTF is used by both VC and EC. In contrast, MC adopts a traditional tolerance factor, which is set to 0.0001.

## 4.2 Incorporating Domain Knowledge

In order to explore the frontier between the feasible and the infeasible region, EC is divided in EC\_f and EC\_inf. The first is composed of feasible solutions and the other of infeasible solutions. Also, we introduce domain knowledge through the mutation operators, which modify the decision variables involved in a particular constraint (either the constraint with the highest violation, or the one with the most negative value, depending if the cell is infeasible or not, respectively).

## 4.3 Mutation Operators

Each population that reacts (EC\_f, EC\_inf and MC) has its own mutation operator. These operators are described next.

The mutation operator for EC\_inf works in the following way: first, it identifies the most violated constraint, say  $c$ . If this constraint value ( $c$ ) is larger than  $\text{sum\_res}$  divided by the total number of constraints, then we change each bit from each decision variable involved in  $c$  with probability 0.05. Otherwise, we change each bit from one decision variable involved in  $c$ , randomly selected, with probability 0.05.

The mutation operator for EC\_f generates two mutated cells, and the best of them passes to the following iteration. This operator works in the following way:

**First operator:** it identifies the constraint with the most negative value (let's keep in mind that this population only has feasible cells), and changes each bit from each decision variable involved in that constraint, with probability 0.05. This operator tries to reduce the distance between the cell and the frontier with the infeasible region.

**Second operator:** it changes each bit from all the decision variables, with probability 0.05. If after applying mutation, a cell becomes feasible, it is inserted in EC\_f according to an elitist selection. Otherwise, if after applying mutation, a cell becomes infeasible, it is inserted in EC\_inf according to an elitist selection.

The mutation operator for MC applies the following equation:

$$x' = x \pm \left( \frac{N(0,1)lu - ll}{1000000gen|const||dv|} \right)^{N(0,2)} \quad (1)$$

where  $x$  and  $x'$  are the original and mutated decision variables, respectively.  $N(0,1)$  and  $N(0,2)$  refer to a randomly generated number, produced with a uniform distribution between  $(0,1)$  and  $(0,2)$ , respectively.  $lu$  and  $ll$  are the upper and lower limits of  $x$ .  $|const|$  refers to the number of constraints of the problem.  $|dv|$  refers to the number of decision variables of the problem and  $gen$  is the current generation number.

#### 4.4 Replacement Mechanisms

The replacement mechanisms are always applied in an elitist way, both within a population and between different populations. They take into account the value of the objective function or the sum of constraints violation, depending on whether the cell is feasible or infeasible, respectively. Additionally, we always consider a feasible cell as better than an infeasible one. Note that before a cell is inserted into another population, it is first evaluated with the tolerance factor of the receptor population.

Therefore, the general structure of our proposed model for constrained problems is the following:

Repeat a predetermined number of times

1. Randomly generate Virgin Cells
2. Calculate DTF's VC
3. Evaluate VC with its own DTF
4. Insert a percentage of Virgin Cells into the Effector Cells population
5. Repeat a predetermined number of times
  - 5.1. Make the Effector Cells React
  - 5.2. Calculate DTF's EC's
  - 5.3. Evaluate ECs with its own DTF

End Repeat

6. Insert a percentage of Effector Cells into the Memory Cells population
7. Repeat a predetermined number of times
  - 7.1. Make the Memory Cells React
  - 7.2. Evaluate MC

End Repeat

End Repeat

The most relevant aspects of our proposed model are the following:

- The fitness of a cell is determined by the value of the objective function and the value for the constrained functions.
- The size of each population is fixed. But, at first, EC<sub>f</sub>, EC<sub>inf</sub> and MC are empty. Step 4 is the responsible for fill EC<sub>f</sub> and EC<sub>inf</sub>. If at the beginning EC<sub>f</sub> can not be filled with feasible cells from VC, the size of EC<sub>f</sub> must to be less than the fixed value for EC<sub>f</sub>, the following applications of step 4 could be completed it. This situation occurs for EC<sub>inf</sub> too, but considering infeasible cells from VC. Step 6 is in charge to complete MC. First are considered the cells from EC<sub>f</sub> and then the cells from EC<sub>inf</sub>, if it is necessary.

- The model returns the best and worst feasible solutions in MC and the mean of the best feasible solution found in each run.
- All the equality constraints are transformed into inequality constraints, using a tolerance factor  $\delta$ :  $|h(\mathbf{x})| - \delta \leq 0$ .
- VC's cells and MC's cell are sorted using the following criterion: the feasible cell whose objective function values are the best are placed first. Then, we place the infeasible cells that have the lowest sum of constraint violation.
- EC\_f's cells are sorted in ascending order based on their objective function values.
- EC\_inf's cells are sorted in ascending order based on their sum of constraint violation.

#### 4.5 Differences Among the Models

After the explanation of our proposed model, we have described the main differences between T-Cell and the models in Section 3. Those models are based on different immunological theories. Clonal Selection is based on the replication of antibodies according to their affinity. The Immune Network Model is a probabilistic approach to idiotypic networks. Negative Selection is based on the principles of self and nonself discrimination that takes place in the immune system. Additionally, Negative Selection and the T-Cell Model are both based on the mechanisms of the T-Cell. However, these models give a completely different treatment to the cells (in the T-Cell Model) and the detectors (in the Negative Selection model). The Negative Selection model tries to detect a change, whereas the T-Cell Model categorizes the T-cells and it uses their phases in order to achieve different goals.

## 5 Experimental Setup

In order to validate our proposed model, we tested it with a benchmark of 19 test functions taken from the specialized literature [13]. The test functions g02, g03, g08 and g12 are maximization problems (for simplicity, these problems were transformed into minimization problems using  $-f(x)$ ) and the rest are minimization problems.

Our results are compared with respect to Stochastic Ranking [14], which is a constraint handling technique representative of the state-of-the-art in the area. Additionally, we also compared our results with respect to the AIS approach reported in [12]. 25 independent runs were performed for each test problem, each consisting of 350,000 fitness function evaluations. We used a population size, for EC and MC, of 20 cells. And for VC we used 100 cells for all the test functions, except for g03 and g11, in which we used only 10 cells. We adopted a 100% and 50% replacement policy for the cells in EC and MC, respectively. All the statistical measures reported are taken only with respect to the runs in which a feasible solution was reached at the end.

## 6 Discussion of Results

Tables 1, 2 and 3 show the results obtained with the AIS proposed in 12, Stochastic Ranking and our T-Cell Model, respectively.

From Table 3, we can see that our model was able to reach the global optimum in 8 test functions (g01, g04, g06, g08, g11, g12, g16 and g18). Additionally, our model reached feasible solutions close to the global optimum in 7 more test functions (g02, g03, g05, g09, g10, g13 and g15) and it found acceptable (but not too close to the global optimum) feasible solutions for the rest of the test functions.

Comparing our T-Cell Model with respect to Stochastic Ranking (see Tables 2 and 3), our T-Cell Model obtained better results in 9 test functions (g02, g03, g04, g06, g10, g11, g14, g16 and g18). Both approaches found similar solutions for g01, g08 and g12. Our proposed model was outperformed in 5 test functions (g05, g07, g09, g13 and g15). With respect to the mean and worst found solutions, our model was outperformed in all functions, except for g02, g04, g06, g16 and g18.

Comparing our T-Cell Model with respect to the AIS proposed in 12 (see Tables 1 and 3), our T-Cell Model obtained better results in 9 test functions (g01, g02, g03, g05, g06, g07, g10, g11 and g13). However, for g05, our model only converged to a feasible solution in 68% of the runs while the AIS from 12 converged to a feasible solution in 90% of the runs. Both approaches found similar solutions for g04, g08 and g12. Finally, our proposed model was outperformed in g09. With respect to the mean and worst found solutions, our model was outperformed only in g07, g09 and g11.

We also conducted an analysis of variance (ANOVA) of the results obtained by our T-Cell Model and of the results obtained by Stochastic Ranking 15. This analysis indicated that the means between the results of the algorithms had

**Table 1.** Results obtained by the AIS proposed in 12. The asterisk (\*) indicates a case in which only 90% of the runs converged to a feasible solution.

Function	Optimum	<i>Best</i>	<i>Mean</i>	<i>Worst</i>	<i>Std.Dev</i>
g01	-15	-14.9874	-14.7264	-12.9171	0.6070
g02	-0.803619	-0.8017	-0.7434	-0.6268	0.0414
g03	-1.0005	-1.0	-1.0	-1.0	0.0000
g04	-30665.5386	-30665.5387	-30665.5386	-30665.5386	0.0000
g05*	5126.4967	5126.9990	5436.1278	6111.1714	300.8854
g06	-6961.81387	-6961.8105	-6961.8065	-6961.7981	0.0027
g07	24.306	24.5059	25.4167	26.4223	0.4637
g08	-0.095825	-0.095825	-0.095825	-0.095825	0.0000
g09	680.63	680.6309	680.6521	680.6965	0.0176
g10	7049.33	7127.9502	8453.7902	12155.1358	1231.3762
g11	0.799	0.75	0.75	0.75	0.0000
g12	-1.0	-1.0	-1.0	-1.0	0.0000
g13	0.05395	0.05466	0.45782	1.49449	0.3790

**Table 2.** Results obtained by Stochastic Ranking [15]

Function	Optimum	<i>Best</i>	<i>Mean</i>	<i>Worst</i>
g01	-15	-15.0	-15.0	-15.0
g02	-0.803619	-0.803	-0.784	-0.734
g03	-1.0005	-1.0	-1.0	-1.0
g04	-30665.539	-30665.539	-30665.480	-30664.216
g05	5126.4967	5126.497	5130.752	5153.757
g06	-6961.81387	-6961.814	-6863.645	-6267.787
g07	24.306	24.310	24.417	24.830
g08	-0.095825	-0.095825	-0.095825	-0.095825
g09	680.63	680.63	680.646	680.697
g10	7049.33	7050.194	7423.434	8867.844
g11	0.799	0.750	0.750	0.751
g12	-1.0	-1.0	-1.0	-1.0
g13	0.05395	0.053	0.061	0.128
g14	-47.7648	-41.551	-41.551	-40.125
g15	961.71502	961.715	961.731	962.008
g16	-1.905155	-1.905	-1.703	-1.587
g17	8853.539	8811.692	8805.99	8559.613
g18	-0.86602	-0.866	-0.786	-0.457
g19	32.655	33.147	34.337	37.477

**Table 3.** Results obtained by our proposed T-Cell Model. The single asterisk (\*) and double asterisk (\*\*) indicate cases in which only 68% and 92% of the runs converged to a feasible solution, respectively.

Function	Optimum	<i>Best</i>	<i>Worst</i>	<i>Mean</i>	<i>Std.Dev</i>
g01	-15.0	-15.0	-15.0	-15.0	0.0
g02	-0.803619	-0.803102	-0.752690	-0.783593	0.013761
g03	-1.0005	-1.00041	-0.984513	-0.998627	0.004208
g04	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0
g05*	5126.4967	5126.4982	5572.0024	5231.7186	143.0598
g06	-6961.81387	-6961.81387	-6961.81387	-6961.81387	0.0
g07	24.3062	24.3503	28.8553	25.3877	1.2839
g08	-0.095825	-0.095825	-0.095825	-0.095825	0.0
g09	680.63	680.63701	680.94299	680.74652	0.078017
g10	7049.24	7086.7891	9592.7752	7955.0428	766.493969
g11	0.7499	0.7499	0.7983	0.7553	0.010717
g12	-1.0	-1.0	-1.0	-1.0	0.0
g13	0.05394	0.054448	0.94019	0.2232	0.25325
g14	-47.7648	-44.7974	-35.6762	-41.0041	2.328270
g15	961.71502	961.72159	972.126254	964.405444	2.575551
g16	-1.905155	-1.905155	-1.905150	-1.905155	0.000001
g17 **	8853.539	8878.387	9206.116	8981.072	97.022811
g18	-0.86602	-0.86602	-0.665288	-0.811528	0.079574
g19	32.655	37.74956696	50.439198	43.997730	3.714058

significant differences except for g01, g03 and g12. The details of the analysis were, however, omitted, due to space restrictions.

We argue that our proposed model is capable of performing an efficient local search over each cell, which allows the model to improve the feasible solutions found. In cases in which no feasible solutions are found in the initial population, the mutation operators applied are capable of reaching the feasible region even when dealing with very small feasible regions.

Although there is clearly room for improvements to our proposed model, we have empirically shown that this approach is able of dealing with a variety of constrained optimization problems (i.e., with both linear and nonlinear constraints, and with both equality and inequality constraints). The benchmark adopted includes test functions with both small and large feasible regions, as well as a disjoint feasible region.

## 7 Conclusions and Future Work

This paper has presented a new AIS model for solving constrained optimization problems in which novel mutation operators are adopted. One of the operators incorporates knowledge of the problem, by modifying the decision variables involved in the most violated constraint. In order to get close to the frontier between the feasible and infeasible regions, it modifies the decision variables involved in the constraint farthest from zero. For some problems, the feasible region is very small, which makes it difficult to find good solutions. For this reason, we were motivated to develop a dynamic tolerance factor. Such a tolerance factor allows to explore regions of the search space that, otherwise, would be unreachable.

Our proposed model was found to be competitive in a well-known benchmark commonly adopted in the specialized literature on constrained evolutionary optimization. The approach was also found to be robust and able to converge to feasible solutions in most cases. Our analysis of the benchmark adopted made us realize that these test functions require small step sizes. Obviously, a lot of work remains to be done in order to improve the quality of the solutions found, so that the approach can be competitive with respect to the algorithms representative of the state-of-the-art in the area. For example, we plan to improve the mutation operators in order to find more quickly the frontier between the feasible and infeasible regions.

## References

1. Yoo, J., Hajela, P.: Immune network modelling in design optimization. In: Corne, D., Dorigo, M., Glover, F. (eds.) *New Ideas in Optimization*, pp. 167–183. McGraw-Hill, London (1999)
2. Smith, A.E., Coit, D.W.: Constraint Handling Techniques—Penalty Functions. In: Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.) *Handbook of Evolutionary Computation*, Oxford University Press and Institute of Physics Publishing (1997)



3. Garrett, S.M.: How do we evaluate artificial immune systems? *Evolutionary Computation* 13, 145–177 (2005)
4. Forrest, S., Perelson, A., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: *IEEE Symposium on Research in Security and Privacy*, pp. 202–212. IEEE Computer Society Press, Los Alamitos (1994)
5. Jerne, N.K.: The immune system. *Scientific American* 229, 52–60 (1973)
6. Hunt, J.E., Cooke, D.E.: An adaptative, distributed learning system based on the immune system. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 2494–2499. IEEE Computer Society Press, Los Alamitos (1995)
7. Ishiguru, A., Uchikawa, Y.W.: Fault diagnosis of plant system using immune network. In: *MFI 1994. Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Las Vegas, Nevada, USA (1994)
8. de Castro, L.N., Von Zuben, F.: Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation* 6, 239–251 (2002)
9. Kelsey, J., Timmis, J.: Immune inspired somatic contiguous hypermutation for function optimisation. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.M., Beyer, H.G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) *GECCO 2003. LNCS*, pp. 207–218. Springer, Heidelberg (2003)
10. Coello Coello, C.A., Cruz-Cortés, N.: Hybridizing a genetic algorithm with an artificial immune system for global optimization. *Engineering Optimization* 36, 607–634 (2004)
11. Luh, G.C., Chueh, H.: Multi-objective optimal design of truss structure with immune algorithm. *Computers and Structures* 82, 829–844 (2004)
12. Cruz Cortés, N., Trejo-Pérez, D., Coello Coello, C.A.: Handling constrained in global optimization using artificial immune system. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) *ICARIS 2005. LNCS*, vol. 3627, pp. 234–247. Springer, Heidelberg (2005)
13. Liang, J., Runarsson, T., Mezura-Montes, E., Clerc, M., Suganthan, P., Coello, C.C., Deb, K.: Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. Technical report, Nanyang Technological University, Singapore (2006)
14. Runarsson, T.P., Yao, X.: Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation* 4, 284–294 (2000)
15. Cagnina, L., Esquivel, S., Coello, C.C.: A bi-population PSO with a shake-mechanism for solving numerical optimization. In: *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, Singapore, IEEE Press, Los Alamitos (2007)

# A Genetic Representation for Dynamic System Qualitative Models on Genetic Programming: A Gene Expression Programming Approach

Ramiro Serrato Paniagua<sup>1</sup>, Juan J. Flores Romero<sup>1</sup>, and Carlos A. Coello Coello<sup>2</sup>

<sup>1</sup> División de Estudios de Posgrado, Facultad de Ingeniería Eléctrica, Universidad Michoacana de San Nicolás de Hidalgo, Morelia 58000, México

<sup>2</sup> CINVESTAV-IPN, Computer Science Department, México City 07360, México  
raspacorp@yahoo.com.mx, juanf@umich.mx, ccoello@cs.cinvestav.mx

**Abstract.** In this work we design a genetic representation and its genetic operators to encode individuals for evolving Dynamic System Models in a Qualitative Differential Equation form, for System Identification. The representation proposed, can be implemented in almost every programming language without the need of complex data structures, this representation gives us the possibility to encode an individual whose phenotype is a Qualitative Differential Equation in QSIM representation. The Evolutionary Computation paradigm we propose for evolving structures like those found in the QSIM representation, is a variation of Genetic Programming called Gene Expression Programming. Our proposal represents an important variation in the multi-gene chromosome structure of Gene Expression Programming at the level of the gene codification structure. This gives us an efficient way of evolving QSIM Qualitative Differential Equations and the basis of an Evolutionary Computation approach to Qualitative System Identification.

**Keywords:** Genetic Representation, Genetic Programming, Gene Expression Programming, Qualitative Reasoning, QSIM, System Identification, Evolutionary Computation.

## 1 Introduction

A Qualitative Model is a qualitative description of a phenomenon, that description is useful for answering particular questions about it; those questions are focused on the phenomenon-behavior's qualitative characteristics. The phenomena could be Dynamic Systems in the real world. QSIM is a formalism that allows us to model qualitatively a Dynamic System, it has a firm mathematical foundation and is easy to understand. In this work QSIM is used for representing Qualitative Models in a novel Genetic Representation. The final goal of this representation is to be part of an Evolutionary Algorithm which will perform System Identification (System Identification is the task of discovering a model from observations [6], also called Model Learning). The proposed Genetic Representation is easy to implement because it does not use any complex data structure (such as nonlinear pointer-based trees or

graphs), it uses a linear chromosome representation based on Gene Expression Programming (GEP) created by Ferreira [4]. Nevertheless, our representation makes important variations which give it the capability of storing Qualitative Differential Equations in QSIM form.

There is few work on Qualitative System Identification using the QSIM representation and Evolutionary Computation. Alen Varsek [2] built a Qualitative Model Learner based on QSIM that uses Genetic Programming (GP). He uses a binary tree representation where the leaves are QSIM constraints and branching points form the hierarchical structure, trees have different sizes and number of leaves in a given interval. The genetic operators used are the classic GP operators described in [3]. The genetic representation is not discussed enough in Varsek's work. He does not describe the Function and Terminal sets used in his GP algorithm. In the examples used in Varsek's paper there are no constraints with corresponding values, it is commented in the article that this assumption is for simplifying the model. However, Varsek does not specify clearly if this choice was made because of a limitation of the representation, a poor performance of the evolution process, an excessive cost in time and space, or any other aspect. Corresponding values give the QSIM constraints an important expressive power; by using them we can introduce pieces of previous knowledge about the system being modeled, and thus more likely to improve the learned-models accuracy.

Another approach on learning Qualitative Models is the work by Khoury, Guerin and Coghil [7]. They built a semi-quantitative model learner using GP, using a tree representation. They do not use QSIM for the model representation but a combination of Fuzzy Vector Envisionment and Differential Planes. By using a Framework called ECJ, they use Automated Defined Functions (ADFs), which allow them to reuse pieces of the tree in any branch of it. The Terminal set contains all the types of leaves in the tree, which can be Ephemeral Random Constants, variables in the form of fuzzy vectors, and finally ADFs that allow the algorithm to embed restrictions. The Function set contains arithmetic operators as well as ADFs. Basically, the tree representation structure is as follows: the root of the tree is a main ADF function, whose number of arguments define the number of branches at the first depth-level in the tree and therefore, the number of constraints encoded. After the first depth level, there are the subtrees that represent the constraints. The trees have not a fixed size, but have a maximum length; they do not describe in detail the genetic operators used; they only comment about the use of crossover and reproduction, so it should be assumed they used the Koza's definitions. In the conclusion they mention that GP is a costly method from the point of view of computational resources, very probably the evaluation of the fitness function is the main factor for that computational cost. Also, another factor could be the nonlinear pointer tree representation if used, the use of a Java based GP framework instead of one based in C/C++ could be another factor.

Section 2 presents the concepts that serve as basis for the development of the present work. Section 3 describes the proposed genetic representation. Section 4 defines the genetic operators that can be applied to the individuals.

## 2 Background

Evolutionary Computation is based fundamentally on Darwin Natural Evolution. Natural Evolution can be seen as a natural population-based optimization process [1], where the stronger individuals are those who are more adapted to their environment and are a product of that optimization process. Those strong individuals have a bigger probability to survive in their environment and as a consequence, to propagate their genetic information through the population in the next generations. In Evolutionary Computation, individuals are possible solutions to problems that human beings want to solve, commonly those problems are engineering or mathematical problems which are not easy to solve using other techniques. Natural evolution does not change the individuals characteristics at the level of their phenotype, it works in their genetics. This indirect change gives this process a powerful exploring mechanism due to the Pleiotropy and Polygeny [1] effects and many others present in the genes coding-decoding process. In Evolutionary Computation there are some paradigms that make use of evolution at the level of genetics, two of them are: Genetic Algorithms and Gene Expression Programming.

One of the main aspects in the evolution process operation is the Genetic Representation. Genetic Representation is the way nature encodes the phenotypic characteristics in the genes, therefore the Genetic Representation must be expressive enough to encode every possible phenotypic characteristic for that specie of individuals. In the same way, Evolutionary-Computation Genetic Representation has to be sufficiently expressive to encode every possible solution for an specific problem in its search space. Another aspect that the Genetic Representation has to satisfy, is the one related to the correct applicability of genetic operations, in Evolutionary Computation it is also desired that genetic operators can be easily implemented.

### 2.1 Genetic Programming

GP is an Evolutionary Computation Paradigm whose aim is to deal with the problem of Program Induction [3]. That is, the discovery, from the search space containing all possible computer programs, of a program that produces some desired output when presented with some particular input. A wide variety of problems can be expressed as problems of program induction. This means that a computer program could be seen as a generic representation form, for possible solutions to those problems. So, computer programs may represent a formula, a control strategy, a video game, a mathematical model, etc. Computer programs are hierarchical structures that can fit a tree form, that tree is called "computer program parse tree", GP population individuals are computer programs parse trees. GP representation is thus a non-linear non-fixed length structure. In this paradigm there is not a clear separation between the phenotype and the genotype; an individual functions simultaneously as genome and phenome.

### 2.2 Gene Expression Programming

GEP is a genotype/phenotype evolutionary algorithm [4]. Its representation is a fixed length multigenic linear chromosome, where the genes have a special structure composed of a head and a tail. It is important to notice that the fixed length affects the

genotype of the individuals, but the decoded individuals (Computer Programs parse trees) can have different sizes and shapes. The GEP individuals encode parse trees of computer programs like in GP but GEP evolutionary process works at the level of the genotype. Ferreira [4] proposes the use of a set of genetic operators: Replication, Mutation, IS Transposition, RIS Transposition, Gene Transposition, 1-Point Recombination, 2-Point Recombination, Gene Recombination (Recombination is also called Crossover). As Ferreira comments [4], the advantages of a Genetic Representation like the one in GEP are the following. The chromosomes are simple entities: linear, compact, relatively small, easy to manipulate genetically. The genetic operators applied to them are less restricted than those used in GP for example, the mutation operator in GP differs from point mutation in nature in order to guarantee the creation of syntactically correct programs (as observed by Ferreira in [4]). The implementation of mutation in GP as shown in [3] first randomly selects a node from the parse tree, then the node and the sub-expression tree below the node are replaced with a randomly generated tree.

The coding in the GEP chromosomes is named Karva language, the genes in the chromosome contain entities called open reading frames (ORFs) whose length defines the length of the sub-expression tree encoded in the gene. The ORFs length could be equal or less than the length of the gene, this allows the possibility of encoding trees with different sizes and shapes. But if the encoded tree is not always using all the space in the gene, what is the function of those non-coding regions? These regions allow the algorithm the modification through genetic operators of the chromosome without restrictions, because the size and structure of the genes remains constant despite the size of the encoded sub-expression tree. Also, these non-coding regions can store important genetic information that can emerge again in the evolutionary process.

### 2.3 QSIM

QSIM is a representation for Qualitative Differential Equations (QDEs); QSIM is also an algorithm for qualitative model simulation. In this paper we will focus on the QSIM representation. QDEs are abstractions of differential equations and differential equations are as well abstractions or models of the real world Physical Systems. QDEs are Qualitative Models, which are general descriptions of the qualitative characteristics and behaviors of a physical phenomenon. These models express states of incomplete knowledge and can be used to infer useful conclusions about the behaviors of that phenomenon.

In the QSIM representation [5], a QDE is a 4-tuple  $\langle V, Q, C, T \rangle$ , where  $V$  is a set of qualitative variables (this variables represent reasonable functions of time);  $Q$  is a set of quantity spaces one for each variable in  $V$ ;  $C$  is a set of constraints applying to the variables in  $V$ ;  $T$  is a set of transitions which define the domain of applicability of the QDE. The quantity space of a variable is a totally ordered list of important values that serve as qualitative-regions boundaries, those values are called landmark values or simply landmarks. The qualitative constraints are relationships among the qualitative variables in the QDE. There is a basic repertoire of constraints in QSIM. Figure 1 lists those constraints and their meanings.

In Figure 1, the points between brackets are the corresponding values (brackets indicate they are optional), which are tuples of landmark values that the variables can take in some constraint; in other words, the point where the constraint is satisfied. In the description of the basic set of constraints shown in Figure1, there are some constraints that do not use corresponding values, these are the derivative and the constant constraints.

(add x y z [(a1 b1 c1) (a2 b2 c2) ...]) iff  $(\forall t) x(t) + y(t) = z(t)$  and  $(\forall i) a_i + b_i = c_i$  {corresponding values}  
 (mult x y z [(a1 b1 c1) (a2 b2 c2) ...]) iff  $(\forall t) x(t) \cdot y(t) = z(t)$  and  $(\forall i) a_i \cdot b_i = c_i$   
 (minus x y [(a1 b1) (a2 b2) ...]) iff  $(\forall t) y(t) = -x(t)$  and  $(\forall i) b_i = -a_i$   
 (d/dt x y) iff  $(\forall t) y(t) = (d/dt) x(t)$   
 (constant x)  
 (M+ x y [(a1 b1) (a2 b2) ...]) iff  $(\forall t) y(t) = f(x(t))$  where f belongs to the set of reasonable monotonously increasing functions and  $(\forall t)(\forall i) x(t) = a_i$  iff  $y(t) = b_i$   
 (M- x y [(a1 b1) (a2 b2) ...]) iff  $(\forall t) y(t) = f(x(t))$  where f belongs to the set of reasonable monotonously decreasing functions and  $(\forall t)(\forall i) x(t) = a_i$  iff  $y(t) = b_i$

**Fig. 1.** The QSIM constraints basic set

### 3 Genetic Representation

Genetic Programming (GP) [3] and Gene Expression Programming (GEP) [4] are Evolutionary Algorithms that evolve computer programs. They differ in the representation and in the form of their genetic operators; both of them use two element sets that form the alphabet used in their representation. Those sets are F the set of functions and T the set of terminals [3][4].

The GEP representation encodes an expression tree, like in GP. The difference between them is that GP individuals are directly the computer-programs parse trees while GEP uses a phenotype-genotype approach, the genotype is structured by a multi-gene chromosome and the phenotype is the computer-program parse tree. Each gene encodes a sub-expression tree (a piece of the computer-program parse tree) using the Karva notation, which is just a width-first linearization of the sub-expression tree (see [4]), the full expression tree (the computer-program parse tree) encoded in the chromosome is the result of linking the sub-expressions in each gene using a link function [4] (i. e. A function used to join the sub-expression trees in the decoding process). Each gene in the GEP Genetic Representation has a two-part structure which is formed by a head and a tail [4], this structure guarantees that every gene decodes to a valid sub-expression tree; the head contains functions or terminals while the tail contains only terminals, see Figure 2.

The name for the proposed representation is "Qualitative Model Genetic Representation" (QMGR).

In GP and GEP there are 2 sets F and T whose union contains all possible elements in an individual tree or chromosome; in QMGR we have F, T and a third one L. The F set is formed of the QSIM constraints; T contains the qualitative variables given by the user; L is the set of all landmark values of the variables quantity spaces in the model. The union of these three sets contains all possible elements in a QMGR chromosome.

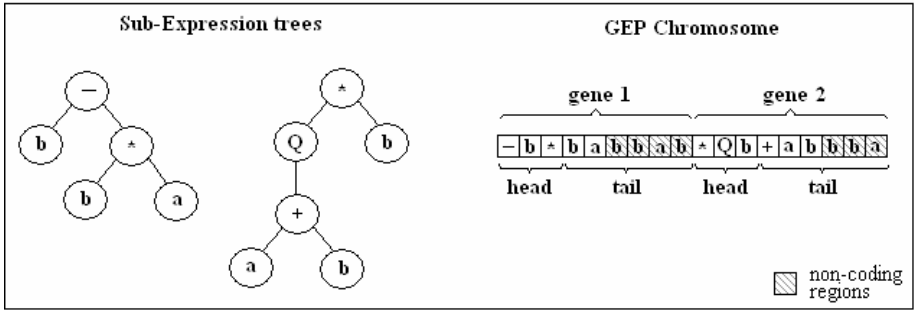


Fig. 2. GEP Genetic Representation

For QSIM models it is not needed to use the Karva notation because those models do not have a tree form necessarily. The structure of the chromosome in the proposed Genetic Representation has a head-tail structure as GEP, we added a third part called CV because it contains the corresponding values of the QSIM constraints. We fixed the head's length to one because it is always encoded one QSIM constraint in a gene (i. e. the head of each gene stores the name of a constraint). The length of the tail is determined by the maximum arity of the functions in  $F$  since the tail contains the arguments applied to the constraint defined in the head; Equation 1 determines the length of the CV.

$$cvLength = \max A * numCV \tag{1}$$

where  $numCV$  is the number of corresponding values to be encoded;  $\max A$  is the greatest arity of all functions in the  $F$  set. Table 1 shows a comparative between GEP and QMGR chromosome's structure.

In QMGR it is encoded one QSIM constraint per gene, therefore the number of constraints in an individual is the same as the number of genes. The user has to make a good choice of the number of genes he would use, as well as the other parameters in the algorithm.

Figure 3 shows how a set of QSIM constraints is encoded in a chromosome using QMGR. For the easy reading and for efficiency in storing the chromosome's linear structure, we use a mapping between the name of a QSIM constraint and a one-character symbol to be stored in the chromosome (this could be also used with the variables and the landmarks if needed) see Figure 3.

This representation provides a generic structure that allows encoding restrictions of any arity and any number of corresponding values, which are encoded in the CV region of each gene. To encode constraints that do not use corresponding values, we need a form of representing the absence of them. In a QMGR individual all genes in a chromosome have the same structure, for example if we have a maximum arity of 3 in the  $F$  set and a number of corresponding values equal to 2, the length of the CV will be 6 by using the equation (1). The CV length is a parameter that affects all the individuals in an instance of the QMGR. We define an instance of QMGR as the set which contains all the individuals being processed or studied, the  $F$ ,  $T$ , and  $L$  sets, as well as the structural characteristics of the chromosomes in that instance, those are:

the head, tail, CV and gene lengths, the number of corresponding values, the number of genes and the chromosome length.

To solve the problem of encoding constraints that do not use corresponding values or that have a less number of them than the given as parameter, we can introduce an element to the L set which represents the absence of a corresponding value. This element could be represented as the symbol “#” and called “null element”. Thus, the chromosome-decoding parser will know when to omit the creation of corresponding values if it finds a null character. This approach allows QMGR to encode constraints that do not use corresponding values or to modify the number of them. This null element can be inserted in the individuals randomly during the initial population creation in an evolutionary algorithm. We also use another approach for dealing with the absence of corresponding values by means of the crossover operator, this approach is described in section 4.1.

## 4 The Genetic Operators

The role of genetic operators in an evolutionary computation algorithm is to serve as the evolutionary-process engine. Genetic operators allow the evolutionary process to explore and exploit the search space. The genetic-operators main goal is the introduction of genetic diversity in the population of individuals in an evolutionary process. We propose three genetic operators for using with QMGR individuals, one- and two- point crossover and mutation.

**Table 1.** GEP and QMGR chromosome’s structure comparative

Structure’s part	GEP	QMGR
Head	Contains symbols from F U T; the length can be defined of any size.	Contains symbols from F; the length is fixed to 1.
Tail	Contains symbols from T; the length is a function of the head’s length and the maximum arity of all functions in F.  $t=h(n-1)+1$ Where t is the length of the head, and n is the maximum arity.	Contains symbols from T; the length is always equal to the maximum arity of the constraints in F.
CV	Not existent.	Contains symbols from L; its length is a function of the number of corresponding values and the maximum arity of the constraints in F, see equation 1.



### 4.1 Crossover

The crossover genetic operator can be easily applied to QMGR individuals. The structure of QMGR allows us to implement the crossover operators defined in GEP and in Genetic Algorithms; this feature can be generalized to n-point crossover. The QMGR structure guarantees that any offspring derived from the crossover operation will encode a valid QSIM-QDE, this is inherited from the GEP representation [4]. The one- and two- point crossover operators are illustrated in Figure 4.

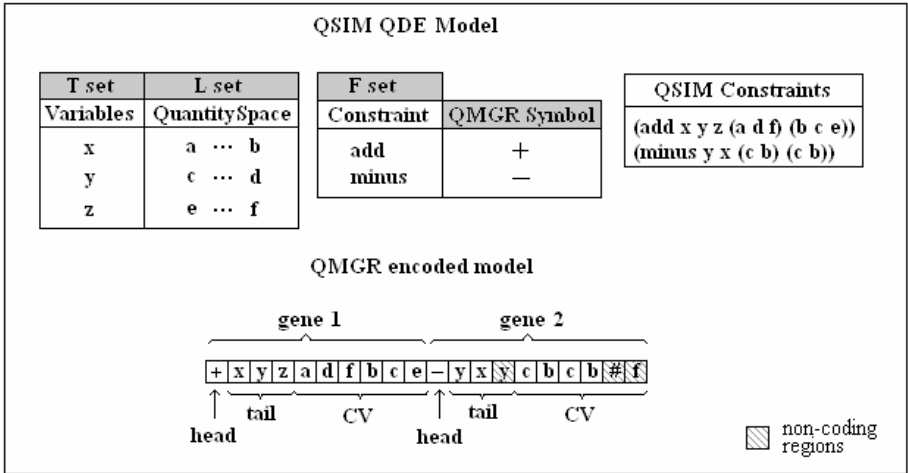


Fig. 3. QMGR Genetic Representation

It is necessary to observe that crossover can generate offspring with non-valid corresponding values. For example, let us suppose we have the following 1-gene individuals in an instance of QMGR, with the following chromosome structural-parameters: CV length equal to 6, maximum arity equal to 3 and number of corresponding values equal to 2.

```
0 1 2 3 4 5 6 7 8 9
+ x y z a c e b d f
- y z z d e c e a c
```

The quantity spaces of the variables are:

- x: a ... b
- y: c ... d
- z: e ... f

The elements in the positions of the CV in each individual gene are valid landmarks in the quantity space of their respective variables. Therefore, all the decoded corresponding values in the CVs will be valid.

If we select a crossover point between position 4 and 5 the offspring will be the following.

```

0 1 2 3 4 5 6 7 8 9
+ x y z a e c e a c
- y z z d c e b d f

```

Analyzing the genes of these offspring it can be seen that in individual one, the two corresponding values are (a, e, c) and (e, a, c). In the first corresponding value, “e” and “c” are landmarks which not belong to the quantity spaces of the respective variables, these variables are “y” and “z”; in the second corresponding value of the first individual, none of the landmarks belong to the quantity spaces of the corresponding variables. Thus both corresponding values of the CV from the first offspring are invalid. The corresponding values of the second individual are (d, c) and (e, b). The first element of the corresponding values for this individual corresponds to the “y” variable and the second element corresponds to the “z” variable. We observe that the first element of the first corresponding value “d” belongs to the quantity space of “y” which is the corresponding variable for this element, but “e” does not satisfy this rule. Neither “c” which is the second element of the first corresponding value nor “b” which is the second element of the second corresponding value, belongs to the quantity space of the “z” variable. Thus both corresponding values in the second offspring are not valid.

In the example, none of the offspring has a valid corresponding value. The problem of generation of invalid corresponding values through crossover, which also means the generation of non-valid individuals, can be seen as an advantage. It can be used as another (see section 3) way of destroying and varying the number of corresponding values in a constraint. To do that, the chromosome-decoding parser must detect when a symbol in the CV does not belong to the quantity space of the corresponding variable, and therefore not decoding it, jumping to the next one, and so on.

The examples in Benjamin Kuipers’ book [5] indicate, that the percentage of constraints that use corresponding values is very small; that means, in general, that when we model a physical system we rarely know the points the functions pass through. The probability assigned to the crossover operators in Genetic Algorithms, GP and GEP is usually high. If we use QMGR in an evolutionary algorithm and use a high probability value for crossover, it will be very likely that this operator will destroy a lot of corresponding values through generations, resulting this in a continuously decreasing percentage of individuals with valid corresponding values. This behavior allows the learned models to be more similar to those in the real world.

## 4.2 Mutation

It is possible to use the point mutation operator on QMGR but we suggest to restrict the operator in the following case: when the chromosome position to be mutated is located in the CV of the genes, this is, stores a corresponding value, it should change the stored value to other that belongs to the set L and also, to the quantity space of the variable referred by this corresponding value. The reason for this restriction is, for conserving the number of valid corresponding-values in that gene. As seen in section

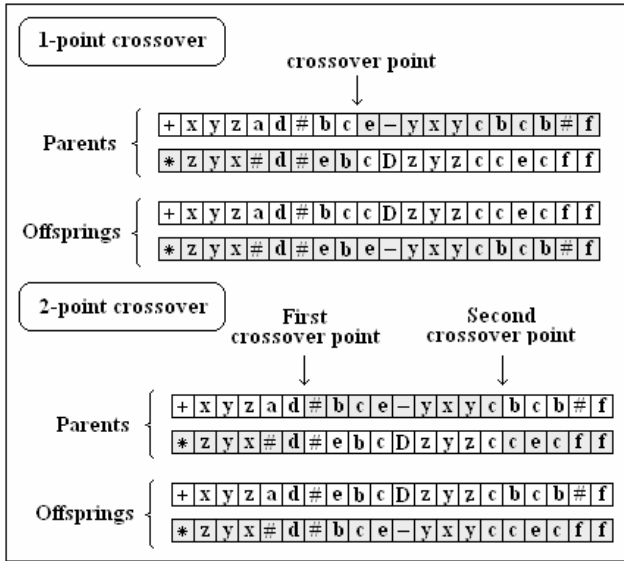


Fig. 4. Crossover operators

4.1 the corresponding values are destroyed when they contain values not existent in the quantity space of the referred variable. Crossover in QMGR has a high corresponding-values destruction power, so it is necessary to avoid that destructiveness in the mutation operator.

There are two more restrictions in the application of the mutation operator, but these ones are mandatory for obtaining valid mutated individuals. When the position to be mutated is located in the tail of any gene, it has to change the stored value only to one element of T. Finally when the position to be mutated is located in the head of any gene, it has to change the stored value only to one element of F.

## 5 Conclusion

In this paper we presented a new Genetic Representation called QMGR. It allows the easy application of genetic operators like crossover and one-point mutation. It uses a linear fixed length multigenic chromosome, which can encode qualitative models of different sizes. QMGR is designed to encode QDEs in the QSIM qualitative representation. QMGR is efficient because it does not need the use of non-linear pointer-based data structures. QMGR's structure encodes the QDEs in a natural form, storing each constraint in one gene in the QMGR chromosomes. This representation can be used in an evolutionary algorithm aggregating a fitness evaluation function. That evaluation function could use the QSIM simulation algorithm for generating the behaviors of the learned models, those behaviors can then be compared to the observations of the system to be modeled. Current research work deals with the implementation of an Evolutionary Algorithm that completes the system identification process at the qualitative level.

**Acknowledgments.** The second author acknowledges support from CONACyT project No. 51729. The third author acknowledges support from CONACyT project No. 45683-Y.

## References

1. Fogel, D.B.: An Introduction to Simulated Evolutionary Optimization. *IEEE Transactions on Neural Networks* 5(1) (January 1994)
2. Varsek, A.: Qualitative Model Evolution. *IJCAI*, 1311–1316 (1991)
3. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge (1992)
4. Ferreira, C.: Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. *Complex Systems* 13(2) (2001)
5. Kuipers, B.: *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge (1994)
6. LJung, L.: *System Identification Theory for the User*. Prentice Hall, USA (1999)
7. Khoury, M., Guerin, F., Coghill, G.M.: Finding semi-quantitative physical models using genetic programming. In: *The 6th annual UK Workshop on Computational Intelligence*, Leeds, 4-6 September, 2006, pp. 245–252 (2006)

# Handling Constraints in Particle Swarm Optimization Using a Small Population Size

Juan C. Fuentes Cabrera and Carlos A. Coello Coello

CINVESTAV-IPN (Evolutionary Computation Group)

Departamento de Computación

Av. IPN No. 2508, San Pedro Zacatenco

México D.F. 07360, México

`jfuentes@computacion.cs.cinvestav.mx`,

`ccoello@cs.cinvestav.mx`

**Abstract.** This paper presents a particle swarm optimizer for solving constrained optimization problems which adopts a very small population size (five particles). The proposed approach uses a reinitialization process for preserving diversity, and does not use a penalty function nor it requires feasible solutions in the initial population. The leader selection scheme adopted is based on the distance of a solution to the feasible region. In addition, a mutation operator is incorporated to improve the exploratory capabilities of the algorithm. The approach is tested with a well-know benchmark commonly adopted to validate constraint-handling approaches for evolutionary algorithms. The results show that the proposed algorithm is competitive with respect to state-of-the-art constraint-handling techniques. The number of fitness function evaluations that the proposed approach requires is almost the same (or lower) than the number required by the techniques of the state-of-the-art in the area.

## 1 Introduction

A wide variety of Evolutionary Algorithms (EAs) have been used to solve different types of optimization problems. However, EAs are unconstrained search techniques and thus require an additional mechanism to incorporate the constraints of a problem into their fitness function [3]. Penalty functions are the most commonly adopted approach to incorporate constraints into an EA. However, penalty functions have several drawbacks, from which the most important are that they require a fine-tuning of the penalty factors (which are problem-dependent), since both under- and over-penalizations may result in an unsuccessful optimization [3].

Particle swarm optimization (PSO) is a population based optimization technique inspired on the movements of a flock of birds or fish. PSO has been successfully applied in a wide of variety of optimization tasks in which it has shown a high convergence rate [10].

This paper is organized as follows. In section 2, we define the problem of our interest and we introduce the Particle Swarm Optimization algorithm. The

previous related work is provided in Section 3. Section 4 describes our approach including the reinitialization process, the constraint-handling mechanism and the mutation operator adopted. In Section 5, we present our experimental setup and the results obtained. Finally, Section 6 presents our conclusions and some possible paths for future research.

## 2 Basic Concepts

We are interested in the general nonlinear programming problem in which we want to:

$$\text{Find } \vec{x} \text{ which optimizes } f(\vec{x}) \quad (1)$$

subject to:

$$g_i(\vec{x}) \leq 0, i = 1, \dots, n \quad (2)$$

$$h_j(\vec{x}) = 0, j = 1, \dots, p \quad (3)$$

where  $\vec{x}$  is the vector of solutions  $\vec{x} = [x_1, x_2, \dots, x_r]^T$ ,  $n$  is the number of inequality constraints and  $p$  is the number of equality constraints (in both cases, constraints could be linear or nonlinear). If we denote with  $F$  to the feasible region (set of points which satisfy the inequality and equality constraints) and with  $S$  to the search space then  $F \subseteq S$ . For an inequality constraint that satisfies  $g_i(\mathbf{x}) = 0$ , the we will say that is **active** at  $\vec{x}$ . All equality constraints  $h_j$  (regardless of the value of  $\vec{x}$  used) are considered active at all points of  $F$ .

### 2.1 Particle Swarm Optimization

Our approach is based in The Particle Swarm Optimization (PSO) algorithm which was introduced by Eberhart and Kennedy in 1995 [4]. PSO is a population-based search algorithm based on the simulation of social behavior of birds within a flock [10]. In PSO, each individual (particle) of the population (swarm) adjusts its trajectory according to its own flying experience and the flying experience of the other particles within its topological neighborhood in the search space. In the PSO algorithm, the population and velocities are randomly initialized at the beginning of the search, and then they are iteratively updated, based on their previous positions and those of each particle's neighbors. Our proposed approach implements equations (4) and (5), proposed in [19] for computing the velocity and the position of a particle.

$$v_{id} = w \times v_{id} + c_1 r_1 (pb_{id} - x_{id}) + c_2 r_2 (lb_{id} - x_{id}) \quad (4)$$

$$x_{id} = x_{id} + v_{id} \quad (5)$$

where  $c_1$  and  $c_2$  are both positive constants,  $r_1$  and  $r_2$  are random numbers generated from a uniform distribution in the range  $[0,1]$ ,  $w$  is the inertia weight that is generated in the range  $(0,1]$ .

There are two versions of the PSO algorithm: the **global** version and the **local** version. In the global version, the neighborhood consists of all the particles of the swarm and the best particle of the population is called the “global best” (*gbest*). In contrast, in the local version, the neighborhood is a subset of the population and the best particle of the neighborhood is called “local best” (*lbest*).

### 3 Related Work

When incorporating constraints into the fitness function of an evolutionary algorithm, it is particularly important to maintain diversity in the population and to be able to keep solutions both inside and outside the feasible region [3,14]. Several studies have shown that, despite their popularity, traditional (external) penalty functions, even when used with dynamic penalty factors, tend to have difficulties to deal with highly constrained search spaces and with problems in which the constraints are active in the optimum [3,11,18]. Motivated by this fact, a number of constraint-handling techniques have been proposed for evolutionary algorithms [16,3].

Remarkably, there has been relatively little work related to the incorporation of constraints into the PSO algorithm, despite the fact that most real-world applications have constraints. In previous work, some researchers have assumed the possibility of being able to generate in a random way feasible solutions to feed the population of a PSO algorithm [7,8]. The problem with this approach is that it may have a very high computational cost in some cases. For example, in some of the test functions used in this paper, even the generation of one million of random points was insufficient to produce a single feasible solution. Evidently, such a high computational cost turns out to be prohibitive in real-world applications.

Other approaches are applicable only to certain types of constraints (see for example [17]). Some of them rely on relatively simple mechanisms to select a leader, based on the closeness of a particle to the feasible region, and adopt a mutation operator in order to maintain diversity during the search (see for example [20]). Other approaches rely on carefully designed topologies and diversity mechanisms (see for example [6]). However, a lot of work remains to be done regarding the use of PSO for solving constrained optimization problems. For example, the use of very small population sizes has not been addressed so far in PSO, to the authors’ best knowledge, and this paper precisely aims to explore this venue in the context of constrained optimization.

### 4 Our Proposed Approach

As indicated before, our proposed approach is based on the local version of PSO, since there is evidence that such model is less prone to getting stuck in local minima [10]. Despite the existence of a variety of population topologies [9], we adopt a randomly generated neighborhood topology for our approach. It also uses a reinitialization process in order to maintain diversity in the population, and it

adopts a mutation operator that aims to improve the exploratory capabilities of PSO (see Algorithm 1).

Since our proposed approach uses a very small population size, we called it **Micro-PSO**, since this is analogous to the micro genetic algorithms that have been in use for several years (see for example [12]). Its constraint-handling mechanism, together with its reinitialization process and its mutation operator, are all described next in more detail.

---

**Algorithm 1.** Pseudocode of our proposed Micro-PSO

---

```

begin
  for  $i = 1$  to Number of particles do
    Initialize position and velocity randomly;
    Initialize the neighborhood (randomly);
  end
  cont = 1;
  repeat
    if cont == reinitialization generations number then
      Reinitialization process;
      cont = 1;
    end
    Compute the fitness value  $G(x_i)$ ;
    for  $i = 1$  to Number of particles do
      if  $G(x_i) > G(xpb_i)$  then
        for  $d = 1$  to number of dimensions do
           $xpb_{id} = x_{id}$ ; //  $xpb_i$  is the best position so far;
        end
        end
        Select the local best position in the neighborhood  $lb_i$ ;
        for  $d = 1$  to number of dimensions do
           $w = rand()$ ; // random(0,1);
           $v_{id} = w \times v_{id} + c_1r_1(pbx_{id} - x_{id}) + c_2r_2(lb_{id} - x_{id})$ ;  $x_{id} = x_{id} + v_{id}$ ;
        end
        end
        cont = cont + 1;
        Perform mutation;
      until Maximum number of generations ;
      Report the best solution found.
    end
  
```

---

#### 4.1 Constraint-Handling Mechanism

We adopted the mechanism proposed in [20] for selecting leaders. This mechanism is based both on the feasible solutions and the fitness value of a particle. When two feasible particles are compared, the particle that has the highest fitness value wins. If one of the particles is infeasible and the other one is feasible, then the feasible particle wins. When two infeasible particles are compared, the



particle that has the lowest fitness value wins. The idea is to select leaders that, even when could be infeasible, lie close to the feasible region.

We used equation (6) for assigning fitness to a solution:

$$fit(\vec{x}) = \begin{cases} f_i(\vec{x}) & \text{if feasible} \\ \sum_{j=1}^n g_j(\vec{x}) + \sum_{k=1}^p |h_k(\vec{x})| & \text{otherwise} \end{cases} \quad (6)$$

## 4.2 Reinitialization Process

The use of a small population size accelerates the loss of diversity at each iteration, and therefore, it is uncommon practice to use population sizes that are too small. However, in the genetic algorithms literature, it is known that it is possible, from a theoretical point of view, to use very small population sizes (no more than 5 individuals) if appropriate reinitialization processes are implemented [12]. In this paper, we incorporate one of these reinitialization processes taken from the literature on micro genetic algorithms. Our mechanism is the following: after certain number of iterations (replacement generations), the swarm is sorted based on fitness value, but placing the feasible solutions on top. Then, we replace the  $rp$  particles (replacement particles) by randomly generated particles (position and velocity), but allow the  $rp$  particles to hold their best position (pbest). The idea of mixing evolved and randomly generated particles is to avoid premature convergence.

## 4.3 Mutation Operator

Although the original PSO algorithm had no mutation operator, the addition of such mutation operator is a relatively common practice nowadays. The main motivation for adding this operator is to improve the performance of PSO as an optimizer, and to improve the overall exploratory capabilities of this heuristic [1]. In our proposed approach, we implemented the mutation operator developed by Michalewicz for Genetic Algorithms [15]. It is worth noticing that this mutation operator has been used before in PSO, but in the context of unconstrained multimodal optimization [5]. This operator varies the magnitude added or subtracted to a solution during the actual mutation, depending on the current iteration number (at the beginning of the search, large changes are allowed, and they become very small towards the end of the search). We apply the mutation operator in the particle's position, for all of its dimensions:

$$x_{id} = \begin{cases} x_{id} + \Delta(t, UB - x_{id}) & \text{if } R = 0 \\ x_{id} - \Delta(t, x_{id} - LB) & \text{if } R = 1 \end{cases} \quad (7)$$

where  $t$  is the current iteration number,  $UB$  is the upper bound on the value of the particle dimension,  $LB$  is the lower bound on the particle dimension value,  $R$  is a randomly generated bit (zero and one both have a 50% probability of being generated) and  $\delta(t, y)$  returns a value in the range  $[0, y]$ .  $\delta(t, y)$  is defined by:

$$\Delta(t, y) = y * (1 - r^{1-(\frac{t}{T})^b}) \quad (8)$$

where  $r$  is a random number generated from a uniform distribution in the range $[0,1]$ ,  $T$  is the maximum number of iterations and  $b$  is a tunable parameter that defines the non-uniformity level of the operator. In this approach, the  $b$  parameter is set to 5 as suggested in [15].

## 5 Experiments and Results

For evaluating the performance of our proposed approach, we used the thirteen test functions described in [18]. These test functions contain characteristics that make them difficult to solve using evolutionary algorithms. We performed fifty independent runs for each test function and we compared our results with respect to three algorithms representative of the state-of-the-art in the area: Stochastic Ranking (SR) [18], the Simple Multimembered Evolution Strategy (SMES) [14], and the Constraint-Handling Mechanism for PSO (CHM-PSO) [20]. Stochastic Ranking uses a multimembered evolution strategy with a static penalty function and a selection based on a stochastic ranking process, in which the stochastic

**Table 1.** Results obtained by our Micro-PSO over 50 independent runs

Statistical Results of our Micro-PSO						
TF	Optimal	Best	Mean	Median	Worst	St.Dev.
g01	-15.000	-15.0001	-13.2734	-13.0001	-9.7012	1.41E+00
g02	0.803619	0.803620	0.777143	0.778481	0.711603	1.91E-02
g03	1.000	1.0004	0.9936	1.0004	0.6674	4.71E-02
g04	-30665.539	-30665.5398	-30665.5397	-30665.5398	-30665.5338	6.83E-04
g05	5126.4981	5126.6467	5495.2389	5261.7675	6272.7423	4.05E+02
g06	-6961.81388	-6961.8371	-6961.8370	-6961.8371	-6961.8355	2.61E-04
g07	24.3062	24.3278	24.6996	24.6455	25.2962	2.52E-01
g08	0.095825	0.095825	0.095825	0.095825	0.095825	0.00E+00
g09	680.630	680.6307	680.6391	680.6378	680.6671	6.68E-03
g10	7049.250	7090.4524	7747.6298	7557.4314	10533.6658	5.52E+02
g11	0.750	0.7499	0.7673	0.7499	0.9925	6.00E-02
g12	1.000	1.0000	1.0000	1.0000	1.0000	0.00E+00
g13	0.05395	0.05941	0.81335	0.90953	2.44415	3.81E-01

**Table 2.** Comparison of the population size, the tolerance value and the number of evaluations of the objective function of our approach with respect to SR, CHM-PSO and SMES

Constraint-Handling Technique	Population Size	Tolerance Value ( $\epsilon$ )	Number of Evaluations of the Objective Function
SR	200	0.001	350,000
CHMPSO	40	-	340,000
SMES	100	0.0004	240,000
Micro-PSO	5	0.0001	240,000

**Table 3.** Comparison of our approach with respect the Stochastic Ranking (SR)

TF	Optimal	Best Result		Mean Result		Worst Result	
		Micro-PSO	SR	Micro-PSO	SR	Micro-PSO	SR
g01	-15.0000	-15.0001	-15.000	-13.2734	-15.000	-9.7012	-15.000
g02	0.803619	0.803620	0.803515	0.777143	0.781975	0.711603	0.726288
g03	1.0000	1.0004	1.000	0.9936	1.000	0.6674	1.000
g04	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539	-30665.534	-30665.539
g05	5126.4981	5126.6467	5126.497	5495.2389	5128.881	6272.7423	5142.472
g06	-6961.81388	-6961.8371	-6961.814	-6961.8370	-6875.940	-6961.8355	-6350.262
g07	24.3062	24.3278	24.307	24.6996	24.374	25.2962	24.642
g08	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825
g09	680.630	680.6307	680.630	680.6391	680.656	680.6671	680.763
g10	7049.250	7090.4524	7054.316	7747.6298	7559.192	10533.6658	8835.655
g11	0.750	0.7499	0.750	0.7673	0.750	0.9925	0.750
g12	1.000	1.0000	1.000	1.0000	1.000	1.0000	1.000
g13	0.05395	0.05941	0.053957	0.81335	0.067543	2.44415	0.216915

**Table 4.** Comparison of our approach with respect the Constraint-Handling Mechanism for PSO (CHM-PSO)

TF	Optimal	Best Result		Mean Result		Worst Result	
		Micro-PSO	CHM-PSO	Micro-PSO	CHM-PSO	Micro-PSO	CHM-PSO
g01	-15.0000	-15.0001	-15.000	-13.2734	-15.000	-9.7012	-15.000
g02	0.803619	0.803620	0.803432	0.777143	0.790406	0.711603	0.755039
g03	1.000	1.0004	1.004720	0.9936	1.003814	0.6674	1.000249
g04	-30665.539	-30665.539	-30665.500	-30665.539	-30665.500	-30665.534	-30665.500
g05	5126.4981	5126.6467	5126.6400	5495.2389	5461.08133	6272.7423	6104.7500
g06	-6961.8138	-6961.837	-6961.810	-6961.837	-6961.810	-6961.835	-6961.810
g07	24.3062	24.3278	24.3511	24.6996	24.35577	25.2962	27.3168
g08	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825
g09	680.6300	680.6307	680.638	680.6391	680.85239	680.6671	681.553
g10	7049.2500	7090.4524	7057.5900	7747.6298	7560.04785	10533.6658	8104.3100
g11	0.7500	0.7499	0.7499	0.7673	0.7501	0.9925	0.75288
g12	1.0000	1.0000	1.000	1.0000	1.000	1.0000	1.000
g13	0.05395	0.05941	0.068665	0.81335	1.71642	2.44415	13.6695

component allows infeasible solutions to be given priority a few times during the selection process. The idea is to balance the influence of the objective and penalty function. This approach requires a user-defined parameter called  $Pf$  which determines this balance [18]. The Simple Multimembered Evolution Strategy (SMES) is based on a  $(\mu + \lambda)$  evolution strategy, and it has three main mechanisms: a diversity mechanism, a combined recombination operator and a dynamic step size that defines the smoothness of the movements performed by the evolution strategy [14].

**Table 5.** Comparison of our approach with respect the Simple Multimembered Evolution Strategy (SMES)

TF	Optimal	Best Result		Mean Result		Worst Result	
		Micro-PSO	SMES	Micro-PSO	SMES	Micro-PSO	SMES
g01	-15.0000	-15.0001	-15.000	-13.2734	-15.000	-9.7012	-15.000
g02	0.803619	0.803620	0.803601	0.777143	0.785238	0.711603	0.751322
g03	1.0000	1.0004	1.000	0.9936	1.000	0.6674	1.000
g04	-30665.5390	-30665.5398	-30665.539	-30665.5397	-30665.539	-30665.5338	-30665.539
g05	5126.4980	5126.6467	5126.599	5495.2389	5174.492	6272.7423	5304.167
g06	-6961.8140	-6961.8371	-6961.814	-6961.8370	-6961.284	-6961.8355	-6952.482
g07	24.3062	24.3278	24.327	24.6996	24.475	25.2962	24.843
g08	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825
g09	680.6300	680.6307	680.632	680.6391	680.643	680.6671	680.719
g10	7049.2500	7090.4524	7051.903	7747.6298	7253.047	10533.6658	7638.366
g11	0.7500	0.7499	0.75	0.7673	0.75	0.9925	0.75
g12	1.0000	1.0000	1.000	1.0000	1.000	1.0000	1.000
g13	0.05395	0.05941	0.053986	0.81335	0.166385	2.44415	0.468294

In all our experiments, we adopted the following parameters:

- $W$  = random number from a uniform distribution in the range  $[0,1]$ .
- $C1 = C2 = 1.8$ .
- population size = 5 particles;
- number of generations = 48,000;
- number of replacement generations = 100;
- number of replacement particles = 2;
- mutation percent = 0.1;
- $\epsilon = 0.0001$ , except for g04, g05, g06, g07, in which we used  $\epsilon = 0.00001$ .

The statistical results of our Micro-PSO are summarized in Table 1. Our approach was able to find the global optimum in five test functions (g02, g04, g06, g08, g09, g12) and it found solutions very close to the global optimum in the remaining test functions, with the exception of g10. We compare the population size, the tolerance value and the number of evaluations of the objective function of our approach with respect to SR, CHM-PSO, and SMES in Table 2.

When comparing our approach with respect to SR, we can see that ours found a better solution for g02 and similar results in other eleven problems, except for g10, in which our approach does not perform well (see Table 3). Note however that SR performs 350,000 objective function evaluations and our approach only performs 240,000.

Compared with respect to CHM-PSO, our approach found a better solution for g02, g03, g04, g06, g07, g09, and g13 and the same or similar results in other five problems (except for g10) (see Table 4). Note again that CMPSO performs 340,000 objective function evaluations, and our approach performs only 240,000. It is also worth indicating that CHMOPSO is one of the best PSO-based constraint-handling methods known to date.

When comparing our proposed approach against the SMES, ours found better solutions for g02 and g09 and the same or similar results in other ten problems, except for g10 (see Table 5). Both approaches performed 240,000 objective function evaluations.

## 6 Conclusions and Future Work

We have proposed the use of a PSO algorithm with a very small population size (only five particles) for solving constrained optimization problems. The proposed technique adopts a constraint-handling mechanism during the selection of leaders, it performs a reinitialization process and it implements a mutation operator. The proposed approach is easy to implement, since its main additions are a sorting and a reinitialization process. The computational cost (measured in terms of the number of evaluation of the objective function) that our Micro-PSO requires is almost the same (or lower) than the number required by the techniques with respect to which it was compared, which are representative of the state-of-the-art in the area. The results obtained show that our approach is competitive and could then, be a viable alternative for using PSO for solving constrained optimization problems.

As part of our future work, we are interested in comparing our approach with the extended benchmark for constrained evolutionary optimization [13]. We are also interested in comparing our results with respect to more recent constraint-handling methods that use PSO as their search engine (see for example [6]). We will also study the sensitivity of our Micro-PSO to its parameters, aiming to find a set of parameters (or a self-adaptation mechanism) that improves its robustness (i.e., that reduces the standard deviations obtained). Finally, we are also interested in experimenting with other neighborhood topologies and other types of reinitialization processes, since they could improve our algorithm's convergence capabilities (i.e., we could reduce the number of objective function evaluations performed) as well as the quality of the results achieved.

## Acknowledgements

The first author gratefully acknowledges support from CONACyT to pursue graduate studies at CINVESTAV-IPN's Computer Science Department. The second author acknowledges support from CONACyT project No. 45683-Y.

## References

1. Andrews, P.S.: An investigation into mutation operators for particle swarm optimization. In: CEC 2006. Proceedings of the 2006 IEEE Congress on Evolutionary Computation, Vancouver, Canada, July, pp. 3789–3796 (2006)
2. Coello, C.A.C., Pulido, G.T.: Multiobjective optimization using a micro-genetic algorithm. In: Spector, L., Goodman, E.D., Wu, A., Langdon, W., Voigt, H.M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M., Burke, E. (eds.) GECCO 2001. Genetic and Evolutionary Computation Conference, pp. 274–282. Morgan Kaufmann, San Francisco (2001)

3. Coello, C.A.C.: Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* 191(11–12), 1245–1287 (2002)
4. Eberhart, R., Kennedy, J.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948. IEEE Computer Society Press, Los Alamitos (1995)
5. Esquivel, S.C., Coello Coello, C.A.: On the use of particle swarm optimization with multimodal functions. In: *CEC 2003. Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, pp. 1130–1136. IEEE Computer Society Press, Los Alamitos (2003)
6. Aguirre, A.H., Muñoz Zavala, A.E., Villa Diharce, E., Botello Rionda, S.: COPSO: Constrained Optimization via PSO algorithm. Technical Report I-07-04, Center of Research in Mathematics (CIMAT), Guanajuato, México (2007)
7. Hu, X., Eberhart, R.: Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization. In: *SCI 2002. Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, USA IIIS*, vol. 5(July 2002)
8. Hu, X., Eberhart, R.C., Shi, Y.: Engineering Optimization with Particle Swarm. In: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA*, pp. 53–57. IEEE Service Center (April 2003)
9. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. In: *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1671–1676. IEEE Press, Los Alamitos (2002)
10. Kennedy, J., Eberhart, R.C.: *Swarm Intelligence*. Morgan Kauffmann Publishers, San Francisco (2001)
11. Koziel, S., Michalewicz, Z.: Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation* 7(1), 19–44 (1999)
12. Krishnakumar, K.: Micro-genetic algorithms for stationary and non-stationary function optimization. In: *SPIE. Proceedings: Intelligent Control and Adaptive Systems*, vol. 1196, pp. 289–296 (1989)
13. Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello Coello, C.A., Deb, K.: Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. Technical report, Nanyang Technological University, Singapore (2006)
14. Mezura-Montes, E., Coello Coello, C.A.: A simple multimembered evolution strategy to solve constrained optimization problems. *Transactions on Evolutionary Computation* 9(1), 1–17 (2005)
15. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Heidelberg (1996)
16. Michalewicz, Z., Schoenauer, M.: Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation* 4(1), 1–32 (1996)
17. Paquet, U., Engelbrecht, A.: A New Particle Swarm Optimiser for Linearly Constrained Optimization. In: *CEC 2003. Proceedings of the Congress on Evolutionary Computation 2003, Piscataway, New Jersey*, vol. 1, pp. 227–233. IEEE Service Center, Canberra, Australia (2003)
18. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* 4(3), 248–249 (2000)

19. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE Congress on Evolutionary Computation, pp. 69–73. IEEE Computer Society Press, Los Alamitos (1998)
20. Pulido, G.T., Coello Coello, C.A.: A constraint-handling mechanism for particle swarm optimization. In: CEC 2004. Proceedings of the 2004 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1396–1403. IEEE Press, Los Alamitos (2004)

# Collective Methods on Flock Traffic Navigation Based on Negotiation

Carlos Astengo-Noguez<sup>1</sup> and Gildardo Sánchez-Ante<sup>2</sup>

<sup>1</sup> ITESM Campus Monterrey  
castengo@itesm.mx

<sup>2</sup> ITESM Campus Guadalajara  
gildardo@itesm.mx

**Abstract.** Flock traffic navigation based on negotiation (FTN) is a new approach for solving traffic congestion problems in big cities. Early works suppose a navigation path based on a bone-structure made by initial, ending and geometrical intersection points of two agents and their rational negotiations. In this paper we present original methods based on clustering analysis to allow other agents to enter or abandon flocks according to their own self interests.

**Keywords:** MultiAgent Systems, Negotiation, Flock, Clustering.

## 1 Introduction

In the last years, approaches based in biology have been proposed in computer science to deal with complex problems which cannot be solved by more traditional methods. Genetic algorithms, neural networks and ant-systems are just a few to be mentioned.

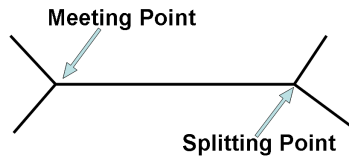
With the increasing number of vehicles being incorporated to the traffic in big cities and the pollution generated by them, it seems like a good idea to look for alternative ways of controlling the flow of vehicles. In this work, we propose a novel approach which we believe could expedite the motion of vehicles, reducing the time to travel as well as the pollution due to car-emissions.

Our work is based on combining ideas coming from Artificial Intelligence, Multiagent Technologies, Mathematics and Nature-based algorithms. This approach is called Flock Traffic Navigation (FTN). In FTN, vehicles can navigate automatically in groups called “flocks” [1]. Birds, mammals and fish seem to organize themselves effortlessly by traveling in this way [2]. A flock consists of a group of discrete objects moving together. In the case of bird flocks, evidence indicates that there is no centralized coordination, but a distributed control mechanism. Navigating in flocks is not only a beautiful manifestation of nature, but a good way of gaining more “collective intelligence” to chose from [3]. People who spend a lot of time looking at European starlings returning to their roost in California, have noticed that a small flock will get lost more frequently, than a big flock [4].



Regarding automated traffic control systems, most of them are based on controlling the flow of individual vehicles -mainly by lights. In contrast, flock traffic navigation allows the coordination of intersections at the flock level, instead of at the individual vehicle level, making it simpler and far safer. To handle flock formation, coordination mechanisms are borrowed from multiagent systems.

The mechanism to negotiate [1] starts with an agent who wants to reach its destination. The agent knows a *a priori* estimation of the travel time that takes to reach its goal from its actual position if he travels alone. In order to win a speed bonus (social bonus) he must agree with other agents (neighbors) to travel together at least for a while. The point in which the two agents agree to get together is called the meeting point, and the point where the two agents will separate is called the splitting point. Together, they form the called bone structure diagram shown in Fig. 1.



**Fig. 1.** Bone Structure Diagram. Two agents agree to meet in their geometrical intersection according to its metric. Before the meeting point and after the meeting point agents will travel at their maximum individual speed. Between the meeting and splitting points agents can add a social velocity bonus.

Individual reasoning plays the main role of this approach. Each agent must compare its *a priori* travel time estimation versus the new travel time estimation based on the bone-diagram and the social bonus, and then make a rational decision.

The decision will be taken considering whether they are in Nash equilibrium or if they are in a Pareto set. In the first case, for any two of them there is no incentive to choose another neighbor agent than the agreed one [5].

If both agents are in Nash equilibrium, they can travel together as partners and they can be benefited with the social bonus. In this moment a new virtual agent is created in order to negotiate with future candidates [6]. An agent in a Pareto set can be added to this bone diagram if its addition benefits it without making any of the original partners to worse off.

Simulations indicate that flock navigation of autonomous vehicles could substantially save time to users and let traffic flow faster [1]. With this approach the new agents in the Pareto set who want to be added into the bone diagram have to share the same original splitting point calculated with the first agents who were in Nash equilibrium.

New problems could arise when using this algorithm. Some of them are listed:

- Can newly incorporated vehicles goals affect the splitting point in the bone diagram in order to improve their agents individual benefit?

- Can other agents that have not been considered in the original Pareto set be incorporated as long as the flock is traveling using rational negotiation with the virtual agent?

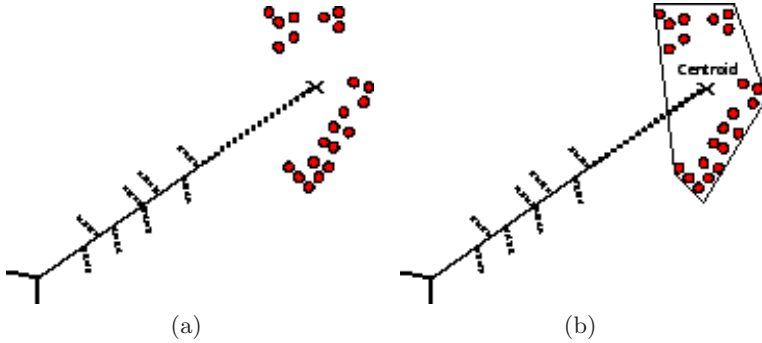
We intend to solve those problems by improving the mechanism proposed in [1] in the context of rational decision making.

## 2 Splitting Point Extension

Suppose that two agents agree to travel together because they are in Nash equilibrium. Then, a virtual agent called the flock spirit will be created for future negotiations. Suppose that there are several others agents in a Pareto set with destinations not so far from the original partners. From the original FTN algorithm, these agents can negotiate with the flock spirit using the algorithm presented in [1]. This approach does not consider the possibility of negotiation with agents encountered later on. Negotiation among flocks at intersections can be done using Dresner and Stone reservation algorithm [7].

### 2.1 Centroids

The geometrical center of a body is known as its centroid. If the body has uniform density, the centroid is also the centre of gravity. In  $\mathbb{R}^2$ , the final destination of flock members can be enclosed by a convex hull. See Fig. 2.



**Fig. 2.** Ending points can be viewed as parts of a convex figure that contains them. The centroid or barycenter can be defined as the center of gravity of a figure.

For  $\mathbb{R}^2$ , the mass of a sheet with surface density function  $\sigma(x, y)$  is given by:

$$M = \iint \sigma(x, y) dA \quad (1)$$

and the coordinates of the centroid:

$$\bar{x} = \frac{\iint x\sigma(x, y) dA}{M} \quad (2)$$

$$\bar{y} = \frac{\int \int y \sigma(x, y) dA}{M} \quad (3)$$

It is well known that the centroid of a convex object always lies in the object. The centroid of a set of  $n$  point masses  $m_i$  is located at positions  $x_i$  is:

$$\bar{x} = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i} \quad (4)$$

If all masses are equal (4) simplifies to

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (5)$$

Based on [8] we can state that the centroid of mass gives the location at which a splitting point should be in order to minimize the distance for each agent goal.

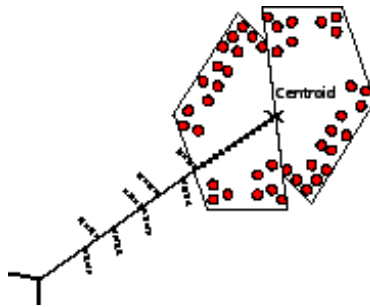
This can be thought as an elegant result but from the multiagent negotiation point of view, agents are not longer in Nash equilibrium so new approaches will be proposed.

## 2.2 Reflections

From the individual point of view, a greedy agent that has agreed to travel with the flock can be tempted to abandon the flock if its goal (ending point) could be reached before the flock splitting point.

One way to solve the problem is to move the meeting point into an earlier position, in order that no agent can be tempted to abandon the flock.

Symmetry results based on reflections in both axes can be applied to solve this problem. The order of complexity for finding the centroid remains as  $O(n)$ .



**Fig. 3.** Centroid based on reflection. Moving the splitting point in to an early position in order that no agent can be tempted to abandon the flock until they reach it.

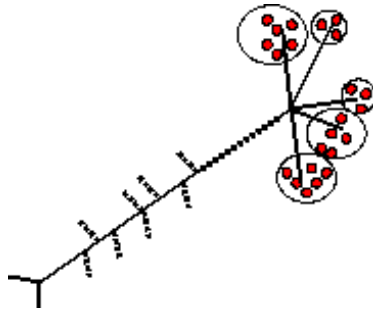
Similar transformations can be applied in order to increase the commitment of the agents to the flock and/or to increase the benefit of cooperation.

### 3 Clustering Techniques

Clustering algorithms are mathematical techniques that partition a data set into subsets (clusters). In other words, clustering is an unsupervised method of classification of patterns (observations, data items or feature vectors) into groups [9].

Cluster analysis is the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity. It is an exploratory data analysis tool which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise [10].

The data in each subset share some common proximity according to some defined metric.



**Fig. 4.** Splitting points clusters. The first splitting point can be found by a centroid reflection method.

A number of methods and algorithms have been proposed for grouping multivariate data into clusters of sampling units. The methods are exploratory and descriptive, and their statistical properties do not seem to be completely developed yet [11].

Typically, cluster methods are highly dependent on the sampling variation and measurement error in the observations. Small perturbations in the data might lead to very different clusters. The choice of the number of clusters may have to be made subjectively and may not follow from the algorithm.

The key concept in the clustering process is the measure of the distances of the observation vectors from one another. Several measures are available, like Euclidean distance, Minkowski distance, Chebychev, Mahalanobis distance or others on the general quadratic kind [11].

$$d^2(x_i, x_j) = (x_i, x_j)^T A^{-1} (x_i, x_j) \quad (6)$$

If we begin with  $N$  objects (agents) the distance may be summarized in the  $N \times N$  symmetric matrix

$$D = \begin{vmatrix} 0 & d_{12} & \dots & d_{1N} \\ d_{12} & 0 & \dots & d_{2N} \\ \dots & \dots & \dots & \dots \\ d_{1N} & d_{2N} & \dots & d_{NN} \end{vmatrix} \quad (7)$$

There are several clustering methods described in the literature as in [8], [9] and [10]. A taxonomy of these methods is presented in [11]. We will describe the ones we believe have a closer relation with our problem.

### 3.1 Single-Linkage Algorithm

The single linkage algorithm combines the original  $N$  single-point clusters hierarchically into one cluster of  $N$  points by scanning the matrix for the smallest distances and grouping the observations into clusters on that basis. Single-linkage algorithm is a Hierarchical clustering method [10].

**Input:**  $N$  travel partners according to [1] negotiation algorithm

**Output:** Matrix of distances  $D$

**foreach** *Pair of agents  $a_i$  and  $a_j$*  **do**

    Calculate their distance  $d_{i,j}$  using the Manhattan metric

$$d_{i,j} = \|x_i - x_j\| + \|y_i - y_j\| \quad (8)$$

    and form matrix  $D$  using equation [7]

**end**

Find the two agents with smallest distance;

Eliminate the rows and columns in  $D$  corresponding to agents  $a_i$  and  $a_j$ ;

Add a row and column of distances for the new cluster according to:

**foreach** *observation  $k$  (agent  $a_k$ )* **do**

$$d_{k(i,j)} = \min(d_{ki}, d_{kj}) \quad (9)$$

$k \neq i, j$

**end**

The result is a new  $(N - 1) \times (N - 1)$  matrix of distances  $D$ . The algorithm continues in this way until all agents have been grouped into a hierarchy of clusters. The hierarchy can be represented by a plot called a dendrogram.

### 3.2 K-Means Algorithm

The K-means algorithm consists of comparing the distances of each observation from the mean vector of each of  $K$  proposed clusters in the sample of  $N$  observations. The observation is assigned to the cluster with nearest mean vector.

The distances are recomputed, and reassignments are made as necessary. This process continues until all observations are in clusters with minimum distances to their means. The typical criterion function used in partitioned clustering techniques is the squared error criteria [8]. The square error for clustering can be defined as

$$e^2 = \sum_{j=1}^K \sum_{i=1}^{N_j} \|x_i^{(j)} - c_j\|^2 \quad (10)$$

Where  $x_i^{(j)}$  is the  $i$ -th agent belonging to the  $j$ -th cluster and  $c_j$  is the centroid of the  $j$ -th cluster.

K-means algorithm is an optimization-based clustering method [10].

```

Input:  $K$  number of clusters and cluster centroids
Output:  $K'$  number of clusters and cluster centroids
foreach new agent  $a_i$  do
  Assign  $a_i$  to the closest cluster centroid;
  while convergence criterion is not met do
    Re-compute the cluster using the current cluster memberships
  end
end

```

Typical convergence criteria are: no (or minimal) reassignment of agents to new cluster centers, or minimal decrease in squared error.

Variations of the algorithm have been reported [11]. Some of them allow splitting and merging of the resulting clusters. Cluster can be split when its variance is above a pre-specified threshold and two clusters are merged when their centroids are below another pre-specified threshold.

### 3.3 Other Algorithms

A successful clustering task depends on a number of factors: collection of data, selection of variables, cleaning of data, choice of similarity measures and finally choice of a clustering method [10].

Density-Based Methods are based on the idea that clusters are high-density regions separated by low density regions.

Grid-Based Methods are based in a projection of the feature space onto a regular grid. Using representative statistics for each cell on the grid is a form of data compression.

Graph-Based Methods are based on a node representation for data points, and pairwise similarities are depicted by the edge weights. Graph partitioning methods can be used to obtain clusters of data points.

Model-Based Methods assume knowledge of a model which prescribes the nature of the data. These methods work appropriately if it is known some *a priori* information [11].

## 4 New Flock Traffic Navigation Algorithm Based on Negotiation

Now, we can propose a new algorithm based on [1] and the previous clustering algorithms 3.1 and 3.2.

Suppose that in time  $t = t_0$  there are  $N$  agents in a neighborhood and each agent makes a broadcast searching for partners according to the distance radius and the algorithm described in [1] and [12].

Suppose agent  $a_i$  and agent  $a_j$  are in Nash equilibrium. Given that each of them can make a rational decision, they agree to travel together using the bone-structure algorithm which includes a meeting and a splitting point, as well as the creation of a virtual agent called the flock spirit.

New agent members (which are in the Pareto set) can be added to the flock if new negotiations between each agent and the flock spirit are made in a rational basis. During the travel time, new negotiations can be done using the original ending-points for each agent for a dynamical clustering procedure using the single-linkage algorithm explained in section 3.1, allowing flock fragmentation and so generating new  $k$  flock spirits.

New agent-partners which do not participate in the earliest negotiations (which means that they are not in the Pareto set) can negotiate with the flock spirit using the centroids of the  $k$  clusters according to the K-means algorithm described in section 3.2.

**Input:** City graph  $G$ , neighboring agents  $L$ , Starting time  $t_0$

**Output:** Splitting points for  $K'$ -groups within the flock

**Part A**,  $t = t_0$ ;

Agents broadcast their actual positions and ending points (goals) and using the bone-diagram algorithm as in [1] find partners that are in Nash equilibrium;

Create the bone structure, a meeting point and a splitting point;

Create a flock spirit;

**Part B**,  $t = t_0 + \delta t$  ;

**foreach** agent in a Pareto Set **do**

    Use the single-linkage algorithm to create new splitting points for each of the  $k$ -clusters;

**end**

Within the flock re-negotiate in order to create new  $k$ -flock spirits;

Calculate the centroid for each cluster;

**Part C**,  $t > t_0 + \delta t$ ;

**foreach** new agent within the broadcast of the flock spirit **do**

    use K-means clustering algorithm for negotiation basis;

**end**

## 5 Concluding Remarks and Future Work

This paper presents an extended algorithm for flock traffic negotiation that allow vehicles to enter in a pre-established negotiation.

Using clustering algorithms allows new coming vehicles to share the social bonus, and gives new chances to reallocate splitting points allowing flock fragmentation. Given that individually rational negotiation has been done according to [4], re-negotiation within flock members can only improve their individual score.

Using the single-linkage clustering algorithm it is clear that new vehicles (agents) incorporated from the Pareto set actually could change the original agreed splitting point leaving  $K$  new splitting points localized at the clusters centroids.

The K-Means clustering algorithm can be used for agents that are not considered in the original Pareto in order to determine if individually they can use the flock bone structure for its personal benefit.

With this new algorithm, flock fragmentation is not only allowed but desirable from the individual rational point of view.

The Dresner and Stone reservation algorithm [7] still works under these considerations because there were not any changes at intersections.

Something that remains to be done is a good set of simulations to better validate the results. So far, we have run a representative, but rather small set of tests.

## References

1. Astengo, C., Brena, R.: Flock traffic navigation based on negotiation. In: ICARA. Proceedings of 3rd International Conference on Autonomous Robots and Agents, Palmerston North, New Zealand, pp. 381–384 (2006)
2. Taylor, C.E., Jefferson, D.R.: Artificial life as a tool for biological inquiry. *Artificial Life* 1(1-2), 1–14 (1994)
3. Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21(4), 25–34 (1987)
4. Toner, J., Tu, Y.: Flocks, herds, and schools: A quantitative theory of flocking. *Phys. Rev. E* 58(4828) (1998)
5. Wooldridge, M.: *Introduction to MultiAgent Systems*. John Wiley and Sons, Chichester (2002)
6. Olfati-Saber, R.: Flocking for multiagent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control* 51(3), 401–420 (2006)
7. Dresner, K., Stone, P.: Multiagent traffic management: an improved intersection control mechanism. In: Proceedings 4th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 471–477. ACM Press, New York (2005)
8. Lewicki, P., Hill, T.: *Statistics: Methods and Applications*. Statsoft (2006)
9. Berkhin, P.: Survey of clustering data mining techniques. Technical report, Accrue Software (2002)
10. Morrison, D.F.: *Multivariate Statistical Methods*, 3rd edn. McGraw-Hill, New York (1990)
11. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. *ACM Computing Surveys* 31, 265–323 (1999)
12. Biswas, S., Tatchikou, R., Dion, F.: Vehicle-to-vehicle wireless communication protocol for enhancing highway traffic safety. *IEEE Communications Magazine* (2006)



# A New Global Optimization Algorithm Inspired by Parliamentary Political Competitions

Ali Borji

School of Cognitive Sciences,  
Institute for Studies in Theoretical Physics and Mathematics, Tehran, Iran  
borji@ipm.ir

**Abstract.** A new numerical optimization algorithm inspired by political competitions during parliamentary head elections is proposed in this paper. Competitive behaviors could be observed in many aspects of human social life. Our proposed algorithm is a stochastic, iterative and population-based global optimization technique like genetic algorithms and particle swarm optimizations. Particularly, our method tries to simulate the intra and inter group competitions in trying to take the control of the parliament. Performance of this method for function optimization over some benchmark multi-dimensional functions, of which global and local minimums are known, is compared with traditional genetic algorithms.

**Keywords:** Function Optimization, Politics, Political Competitions, Evolutionary Algorithms, Political Optimization, Stochastic Optimization.

## 1 Introduction

Global optimization is the task of finding a solution set of an objective function in which it takes its smallest value, the global minimum. From a computational complexity point of view, finding the global minimum of a function is a NP-hard problem, which means that there is in general no algorithm for solving it in polynomial time. Global optimization is a stronger version of the local optimization, in sense that instead of searching for a locally improvable feasible point, it looks for the best point in the search space. In many practical engineering applications finding the globally best point is desirable but not critical, since any sufficiently suboptimal solution which satisfies a specific criterion is acceptable. It is also possible to manually improve it further without optimization in some cases. For such problems there is a little harm in doing an incomplete search [1].

Researchers in different fields have got many inspirations from the solutions that nature has evolved for hard problems. An interesting such areas is optimization. Taking advantage of evolutionary approach nature has selected, genetic algorithms and its variants has been suggested for optimization. Modeling animal behaviors has resulted to particle swarm [2] and ant colony optimization algorithms [3]. Recently simulating social behaviors of humans has led to some solutions in engineering [4], [5].

Our aim in this paper is to introduce a new optimization method inspired from human social behavior in political situations. As we are going to compare our method with genetic algorithms a brief introduction to it is followed.

Genetic algorithm, GA, was first introduced by Holland [6]. It takes the essence of biological evolution by activating mutation and crossing-over among candidates. GA is a stochastic, incomplete, iterative, and population-based global optimization method. In each iteration of GA, a competitive selection weeds out poor solutions. Solutions with high “fitness” are then “recombined” with other solutions by swapping their parts. A “mutation” operation is also applied over solutions by making a small change at a single element. Recombination and mutation are used to generate new solutions that are biased towards regions of the space for which good solutions have already been experienced.

Competitions among political groups during head elections in a parliament have been our main inspiration source for formulating a new method for global optimization in this paper. All individuals of the population are clustered randomly into some groups. Members of a party fall into two categories: regular members and candidate members. Parliamentary head candidates compete in each group to become the final candidate of that group. A final candidate then must compete in next round with candidates of other groups for parliament head position. Intra-party and inter-party competitions guide the algorithm to converge to the global minimum of the objective function. We have overridden some real life details purposely for simplicity. For example in parliamentary system of some countries people vote for candidates both within a group and among groups. This algorithm has some similarities with competitions during athletic championships in which athletics are grouped into teams and they play against each other to ascend from their teams and then play with winners of other teams in the next round. But obviously our method has certain differences. As it can be seen it is also somehow similar to competitions among parties during presidential campaign.

The remainder of this paper is organized as follows. In section two a brief introduction to the political competitions to the extent relevant to our work is described. Details of the proposed optimization method are introduced in section three. It is followed by simulation studies in section four. Finally section five draws conclusions and summarizes the paper.

## 2 Parliamentary Political Competitions

It is a common incident and is repeatedly observed in different aspects of human life that people tend to form social groups. In sociology, a group is usually defined as a collection of humans or animals, who share certain characteristics, interact with one another, accept expectations and obligations as members of the group, and share a common identity. Using this definition, society appears as a large group. Characteristics that members in the group may share include, interests, values, ethnic and linguistic background, and kinship ties. Two types of relationships exist within a group and among groups: competition and cooperation. Members of a group (human social group) have different capabilities making each one suitable for a certain task. Because each member has a collection of needs and capabilities, it forms a potential

for them to cooperate. At the same time they compete to earn higher fraction of group resources. In competition among groups, they compete to get better situations and take the superiority over others to attain the limited sources of their living environment. Although many patterns of such behaviors are observable in human social life, we constrain ourselves to a specific competition-cooperation behavior during parliamentary elections.

A parliamentary system, also known as parliamentarianism is a governmental system in which the power to make and execute laws is held by a parliament. Members of the parliament are elected in general elections by people. People usually vote in favor of parties. Members of a parliament belong to political parties. They support their parties in parliamentary votes. Clustering members of the parliament into clusters, based on the party they belong, results to competitions among parties in trying to gain superiority over other parties. Almost in all democratic countries, political parties form the population of parliaments [7].

There are basically two systems in parliamentary elections: the Majority Election System and the Proportional Representation System. In the majority election system, only one Member of Parliament is elected per constituency. In the proportional representation system several members of parliament are elected per constituency. Basically every political party presents a list of candidates and voters can select a list that is they vote for a political party. Parties are assigned parliamentary seats proportionally to the number of votes they get.

Political parties, either in the parliament or out of it, have members with different levels of power. Those main people of a party try to make good impacts over other regular members with less power. They do that to benefit from their support and votes during elections, etc. Therefore, important members of parties are engaged in competitions and try to find supporters among regular members. On the other hand, regular members have tendency toward more capable persons and usually vote for people they believe in. In this dynamic process, regular members with high capability replace previous candidates. These competitions are among individuals within parties. Another kind of competition is at the level of parties. Political parties compete for gaining more power. Two main goals that parties try to achieve are greater number seats in the parliament and taking the control of government.

### **3 Parliamentary Optimization Algorithm (POA)**

Optimization process in our algorithm is started by first creating a population of individuals. These individuals are assumed to be the members of the parliament. In the next step, population is divided into some political groups (parties) and a fixed number of members with highest fitness are selected as group candidates (leaders).

After partitioning the population, intra-group competition is started. In intra-group competition a regular members get biased toward candidates in proportion to their fitness. It is motivated from the fact that a regular member is usually under impact of superior members. This observation is modeled here as a weighted average of vectors from a regular member to candidates. This causes the evolutionary process search for potential points in search space and provides an exploratory mechanism. At the end of intra-party competition a few candidates with highest fitness are regarded as final

candidates of the group. They compete with candidates of other groups in next stage. Both candidates and regular members of a group are important in determining total power of a group. A linear combination of mean power of candidates and mean power of regular members is considered as the total fitness of a group. As in parliamentary system of some countries no voting mechanism is assumed. Actually, the biasness mechanism could somehow be considered as implicit voting.

Inter-group competition begins just after intra-group competitions ends. Political groups within the parliament perform competition with other groups to impose them their own candidate. In our method, the role of groups is still preserved after introducing a candidate. Each group not being able to compete with others becomes weaker and loses its chance to take the parliament head position.

Groups with a negligible fitness gradually lose their power and ultimately collapse. On the other hand, stronger groups become progressively more powerful and consequently earn higher chance to win the competition. Powerful groups sometimes agree to join and merge into one (at least at on some special occasions) to increase their wining chance. This gives the search mechanism chance to look on more promising areas therefore offers a mechanism for exploitation. The tendency of regular members of a group toward their group candidates along with affinity of powerful groups to join and also collapse mechanism drives convergence to a state in which there exists just one group in the parliament. In contrast to what happens in real world, when algorithm converges, regular members have near the same or equal power as the candidate which is now the head. A step by step description of the algorithm is summarized as follows:

1. Initialize the population.
2. Partition population into some groups.
  - 2.1. Pick  $\theta$  highly fitted individuals as candidates of each group.
3. Intra-group competition
  - 3.1. Bias regular members toward candidates of the group.
  - 3.2. Reassign new candidates.
  - 3.3. Compute power of each group.
4. Inter-group competition
  - 4.1. Pick  $\lambda$  most powerful groups and merge them with probability  $P_m$ .
  - 4.2. Remove  $\gamma$  weak groups with probability  $P_d$ .
5. If stopping condition not met go to 3.
6. Report the best candidate as the solution of the optimization problem.

### 3.1 Population Initialization

A population of initial solutions with size  $N$  is being dispread over the  $d$ -dimensional problem space at random positions. Each individual of the population is coded as a  $d$ -dimensional continuous vector:

$$P = [p_1, p_2, \dots, p_d], p_i \in \mathbb{IR} \quad (1)$$

Each individual could be either a regular member or candidate of a given group. A fitness function  $f$  is used to calculate the strength of an individual.

### 3.2 Population Partitioning

In order to form initial groups, population is portioned into  $M$  divisions. Each group contains  $L$  individuals.  $N$ ,  $M$  and  $L$  are positive integers and are selected in such a way to satisfy the following equation:

$$N = M \times L \tag{2}$$

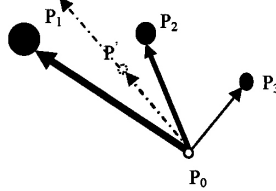
Top  $\theta < L/3$  candidates with high fitness are then considered as candidates of each group. At this point all groups have the same number of members, but in the course of running the algorithm groups might earn different number of individuals because of merge and collapse mechanisms.

### 3.3 Intra-group Competition

Regular members of a group get biased toward candidates after interactions take place between candidates and regular members. This biasness is assumed here to be linearly proportional to weighted average of vectors connecting a member to candidates. Each candidate is weighted to the extent of its candidate fitness as shown in equation 3.

$$p' = p_0 + \eta \left( \frac{(p_1 - p_0) \cdot f(p_1) + (p_2 - p_0) \cdot f(p_2) + (p_3 - p_0) \cdot f(p_3)}{f(p_1) + f(p_2) + f(p_3)} \right) \tag{3}$$

In above formula,  $\eta$  is a random number between 0.5 and 2 and allows the algorithm to search in a local search area around candidates. Another alternative mechanism is to use large values of  $\eta$  at first iterations and the gradually reduce it, perhaps by analyzing variance. Fig. 1 illustrates biasing mechanism.



**Fig. 1.** Biasing a member toward candidates

A regular member is allowed to change, only if it takes a higher fitness value. After biasing, regular members might have higher fitness values than candidates. In such cases, a reassignment of candidates is done. Let  $\underline{Q}^i = [Q_1, Q_2, \dots, Q_\theta]$  be the vector of candidates and  $\underline{R}^i = [R_{\theta+1}, R_{\theta+2}, \dots, R_1]$  the remaining regular members of the  $i$ -th group, power of this group is calculates as:

$$Power^i = \frac{m \times Avg(\underline{Q}^i) + n \times Avg(\underline{R}^i)}{m + n}; m > n \tag{4}$$

### 3.4 Inter-group Competition

Stronger groups sometimes, join and merge to one group in order to amplify their power. To perform merging, a random number is generated and if it is smaller than  $P_m$ ,  $\lambda$  most powerful groups are picked and merged into one. During the course of running algorithm, weak groups are removed to save computation power and reduce function evaluations. Like merging, a random number is generated and if it is smaller than  $P_d$ ,  $\gamma$  groups with minimum power are eliminated.

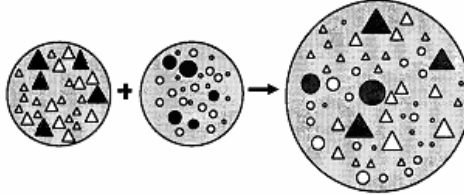


Fig. 2. Merging two groups into one group

### 3.5 Stopping Condition

At the end of algorithm, a group wins the competitions and its best member (candidate with maximum fitness) is considered as the solution of the optimization problem. Two stopping conditions are possible. Algorithm terminates if either maximum number of iterations is reached or during some successive iterations no significant enhancement in fitness is observed.

## 4 Simulation Studies

Several experiments are conducted in this section to demonstrate success of the proposed algorithm. Specifically, capability of the algorithm in finding global minimum of three benchmark functions 'Sphere', 'Rastrigin' and 'Ackley' is investigated. Plots of these functions in two dimensions are shown in Fig. 3. Efficiency of the parliamentary optimization algorithms is also compared with traditional genetic algorithm over these problems. To do experimentation with genetic algorithm, GA toolbox provided with MATLAB® was used. Table 1 shows POA parameter values for minimizing optimization function. In order to do a fair comparison with GA, initial conditions and number of initial individuals were identical in simulations.

### 4.1 Sphere Function

Equation five defines the sphere function in  $n$  dimensions. Dynamics of individual behaviors of the best group around the point with the best fitness is shown in top part of Fig 4. It can be observed that individuals move toward the optimal point from initialization area. In their progress towards the optimal point, individuals are biased toward the candidates of groups. This process gradually moves the points toward the

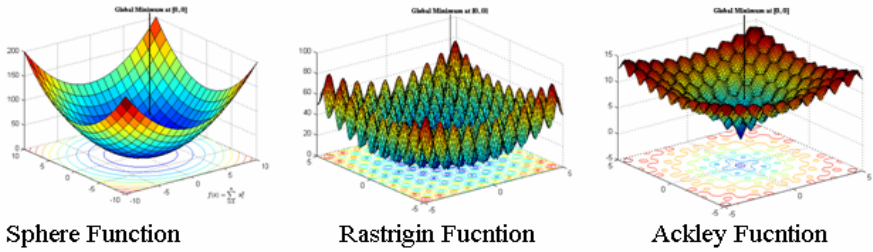


Fig. 3. Plot of functions used in simulations. Black vertical bar points to minimum.

Table 1. POA parameter values for mimization of three optimization problems

Parameter	Quantity	Sphere	Rastrigin	Ackley
N	Number of groups	5	6	10
L	Group size (individuals)	8	10	10
X <sub>ini</sub>	Initial search area	-40 < X <sub>ini</sub> < -30	-20 < X <sub>ini</sub> < -10	-20 < X <sub>ini</sub> < -10
P <sub>m</sub>	Merge probability	0.01	0.05	0.01
P <sub>d</sub>	Deletion probability	0	0.0025	0.002
θ	Number of candidates	2	3	2
d	Dimension	2	2	10
η	Biasness parameter	<b>0.5 &lt; random number &lt; 2</b>		
m	Candidate weighting constant	1	1	1
n	Member weighting constant	0.01	0.01	0.01
λ	Groups to be merged	<b>2</b>		
γ	Groups to be deleted	<b>1</b>		
Itr	Maximum iterations	100	500	1000

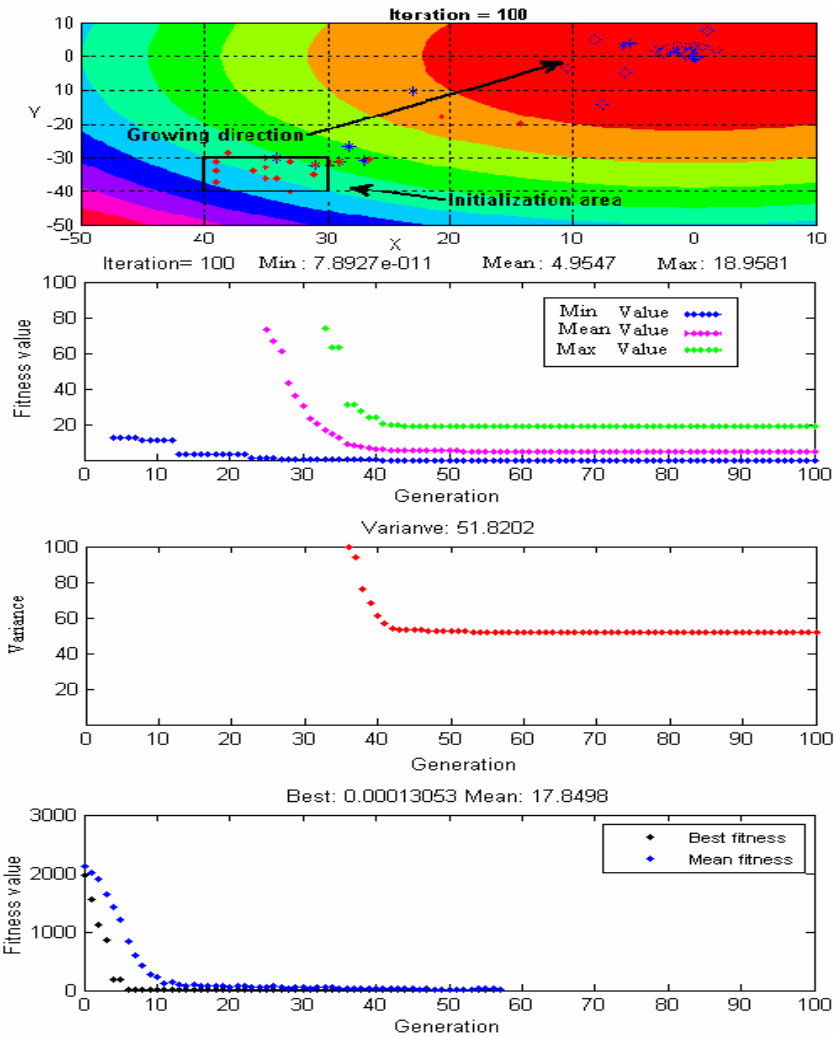
global minimum. The minimum value of the sphere function discovered by POA was  $y = 7.89 \times 10^{-11}$  at the point  $x = [-0.1413e-3, -0.0662e-3]$ . It is known that the optimal value of this function is zero for the point  $[0, 0]$  in the  $x$ - $y$  plane. Minimum, maximum and average of individuals in population as well as fitness variance is plotted in Fig. 4. Compared with GA, our methods showed significance enhancement o 100 iterations.

$$f(x) = \sum_{i=1}^n x_i^2 \tag{5}$$

### 4.2 Rastrigin Function

We address another challenging optimization problem, which is minimization of Rastrigin function to demonstrate the effectiveness the POA. Fig. 3 clearly shows that the Rastrigin function has numerous local minima. However, it has just one global minimum, at the point  $[0, 0]$  in the  $x$ - $y$  plane, as indicated by the vertical line in the plot, where the value of the function is zero. Rastrigin function is defined as below:

$$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \tag{6}$$



**Fig. 4.** From top to bottom: Convergence of the population toward the optimal point over Sphere function. Minimum, mean and maximum fitness are plotted for each generation over entire population. High variance in first iterations decreases at algorithm converges to the solution. Performance of GA over sphere function is significantly worse than POA.

Performance of POA and GA over Rastrigin function is shown in Fig. 5. POA reached absolute zero (in MATLAB) after 293 iterations. Algorithm stopped after no change was seen over fitness landscape. Again in comparison with GA a significant improvement was achieved.

### 4.3 Ackley Function

Ackley function is a challenging and favorite benchmark problem for optimization algorithms shown in equation 7. It could be observed from Fig. 3 that the function has



only one global minima at [0, 0] in x-y plane with also numerous local minima. In this experiment we aimed to compare the efficiency of POA with GA over a high dimensional optimization problem. Results over this function are illustrated in Fig. 6.

$$f(x) = 20 + e - 20e^{-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{-\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)} \quad (7)$$

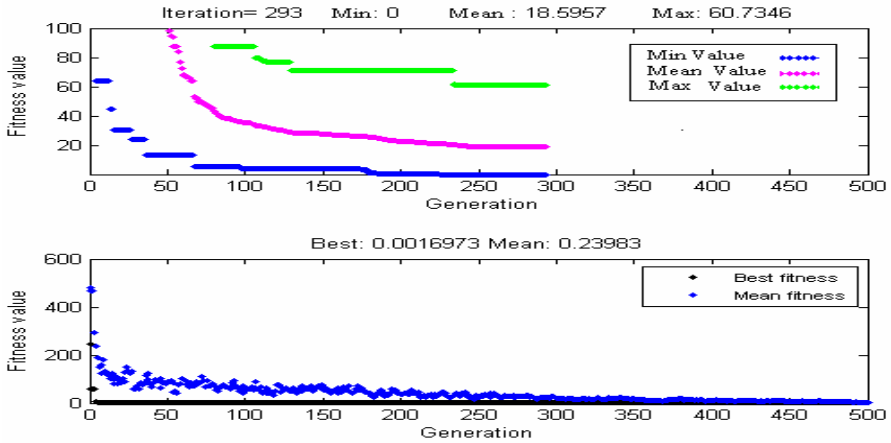


Fig. 5. Performance of POA and GA over Rastrigin function is shown in this figure. Note that GA has fall into local minima.

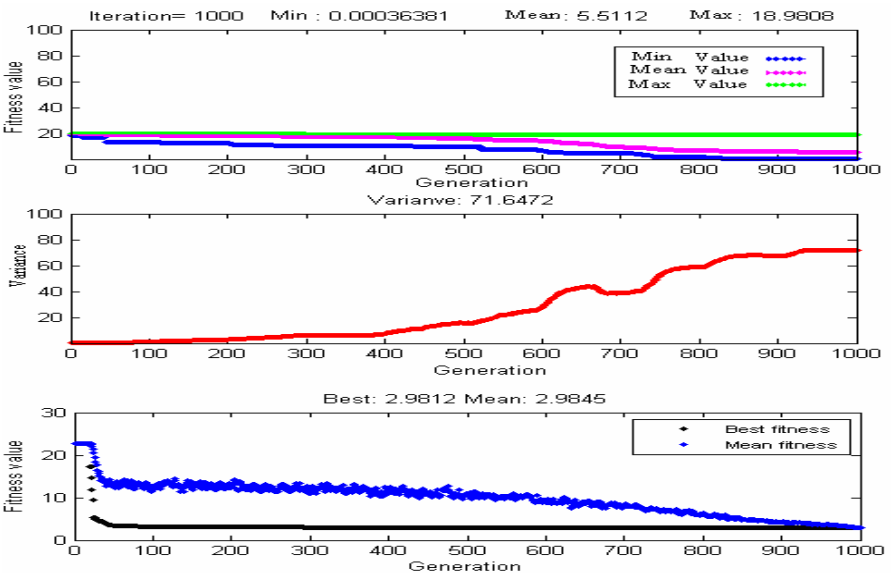


Fig. 6. POA is compared with GA over 10-dimensional Ackley function. Again GA has fall into local minima. Gradual increase in variance is because of local minima in function.

## 5 Conclusions

In this work, we introduced a new global optimization algorithm which is inspired from competitions among political parties trying to take the control of the parliament. Although we have bypassed some of the details of these competitions, proposed algorithm still has acceptable efficiency.

An initial population is created at the first step. It is then partitioned into some political groups. Each member of a given group is either a regular member or a superior member (candidate) of that group. Intra-group competition is the attempt of superior members to get the support of regular members. Regular members fluctuate with their bias toward superior members based on their achievements. In inter-group competition, political groups engage in competition and cooperation behaviors to win a good situation. This cooperation is modeled in this paper as merging those more powerful groups to one bigger more powerful one. Some weak groups which have no positive effect on search process are removed. That way, stronger groups become gradually more powerful while weaker ones become weaker and finally collapse.

At the end of these competitions, the most superior of the most powerful party becomes the leader or the head of the parliament and is considered as the solution of the optimization problem.

Several numerical simulations are carried out to investigate the convergence of POA over three benchmark optimization problems. Results show significant enhancement over staged genetic algorithm over three problems.

As it can be from simulations our proposed optimization algorithm captures important essences of political competitions fairly well and is capable to find desired minima very fast in comparison with other stochastic search algorithms. As an optimization algorithm, it has the additional desirable properties of capability to deal with complex and non-differentiable objective functions and escapes from local optima.

Through investigation of algorithm parameters, comparison with other optimization techniques like particle swarm and ant colony optimization as well as experimenting with a rich repertoire of high dimensional benchmark problems are the areas authors suggest for future works.

## References

1. Horst, R., Pardalos, P.M., Thoai, N.V.: Introduction to Global Optimization, 2nd edn. Kluwer Academic Publishers, Dordrecht (2000)
2. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Computer Society Press, Los Alamitos (1995)
3. Dorigo, M., Maniezzo, V., Colomi, A.: The ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics. Part B. Cybernetics 26(1), 1–13 (1996)

4. Hemingway, C.J.: Socio-cognitive Research: Toward a socio-cognitive theory of information systems: an analysis of key philosophical and conceptual issues. In: the Workshop Proceedings of Information Systems: Current Issues and Future Changes (1998)
5. Sharples, M.: Socio-cognitive engineering: a methodology for the design of humancentred technology. *European Journal of Operational Research* (2002)
6. Holland, J.H.: *Adoption in Natural and Artificial Systems*. University of Michigan, Ann Arbor (1975)
7. Shourie, A.: *The parliamentary system*. Rupa & Co, USA (2007)

# Discovering Promising Regions to Help Global Numerical Optimization Algorithms

Vinícius V. de Melo, Alexandre C. B. Delbem, Dorival L. Pinto Júnior,  
and Fernando M. Federson

State University of São Paulo, São Carlos - SP, Brazil  
{vmelo, acbd, leao}@icmc.usp.br,  
federson@gmail.com

**Abstract.** We have developed an algorithm using a Design of Experiments technique for reduction of search-space in global optimization problems. Our approach is called Domain Optimization Algorithm. This approach can efficiently eliminate search-space regions with low probability of containing a global optimum. The Domain Optimization Algorithm approach is based on eliminating non-promising search-space regions, which are identified using simple models (linear) fitted to the data. Then, we run a global optimization algorithm starting its population inside the promising region. The proposed approach with this heuristic criterion of population initialization has shown relevant results for tests using hard benchmark functions.

## 1 Introduction

A frequent search strategy to work with complex optimization problems have been the exploration of scattered points in the solution space. As there is no information about a global optimum location before solving an optimization problem, algorithms based on such strategy can evenly scan a feasible region of the search-space to determine good solutions (points) for better exploration in subsequent iterations. As the algorithm iterates improving a population (set) of solutions, a subset of solutions move, in general, closer to a global optimum.

In order to reduce the running time of optimization algorithms, advanced metaheuristics have been developed. For example, these approaches can emphasize promising regions of the solution space obtaining better performance. Metaheuristics using complex probabilistic models to determine promising regions have shown relatively high capacity to find optimum solutions. On the other hand, such strategies are computationally less efficient due to the complexity of the probabilistic models employed.

We propose a search strategy using Design of Experiments (DoE) [10] to determine non-promising regions. This approach is iteratively applied to eliminate regions where the algorithm guarantees, with high probability, that there is no global optimum. The algorithm finishes if it cannot guarantee that an additional reduction will not lose the optimum solution. Our approach, a Domain Optimization Algorithm (DOA) belongs to a class of algorithms called Search-space Reduction Algorithms (SRAs) [15,2]. A SRA is a technique to determine one or more promising regions before the use of an

optimization algorithm. This way, one can concentrate the population into the limits defined by the SRA, where there is a high probability of finding the global optimum.

This paper is presented as follows. In section two, the DoE methodology applied in this work is introduced. In section 3, the DOA is presented and briefly commented. In section 4, the applicability of our proposed SRA is demonstrated using some global optimization benchmark functions. Next, we finalize this paper with some conclusions and future work.

## 2 Design of Experiments

One of the objectives of statistical methods is to increase the efficiency of processes, and it is exactly what optimization algorithms, like Genetic Algorithms (GAs) [5], need. There is an interesting relationship between GAs and an area of statistics known as DoE [10]. Some important points about DoE are that it tries to extract information using a minimum set of points, incorporates the discovered information in a cumulative manner, need human interaction and interpretation to take decisions and has an explicit model with which it attempts to account the observed phenomena. That relationship was identified by Reeves and Wright [13][14].

Thus, it is reasonable to imagine that one ideal algorithm should be capable of automating the decisions that must be taken in a typical DoE to guide the search process (using a GA, for example) to evaluate the next best point, chosen according to the whole information available. This strategy was adopted in the Orthogonal Crossover [8] and Taguchi Crossover [1] operators, with great success.

In this work, we use the information obtained by a DoE technique to select a limited region of the search-space where the best point (global optimum) can be located, called a promising region. Concentrating the search in this promising region, one can expect a reduction of the number of evaluated points. To select the promising region, our approach uses the Multiple Linear Regression technique, explained in the following section.

### 2.1 Multiple Linear Regression

Multiple Regression is a set of statistical methods used to build mathematical models when one intends to study the behavior of a response variable according to one or more explicative/predictive variables of a determined process. As such model is subtle to errors, it is common to accept models with reasonable precision. The regression analysis is a technique applied in several fields of knowledge [9].

**Regression Hypothesis Tests.** In the regression analysis, it is important to evaluate how much the relationship among the response value ( $y$ ) and the explicative variables ( $x$ ) can be considered significant. This can be made by hypothesis tests, as presented by Myers and Montgomery [11].

- Significance Tests for Regression

This test verifies if there is any influence of the explicative variables upon the response variable. The hypothesis are:

$$H_0 : \hat{\beta}_1 = \dots = \hat{\beta}_k = 0$$

$$H_1 : \hat{\beta}_j \neq 0, j = 0, \dots, k, \text{ for at least one } j$$

The estimatives  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_k$  of the coefficients  $\beta_0, \beta_1, \dots, \beta_k$  of the linear model can be calculated by the Minimum Square Method [69], where  $k$  is the number of factors/variables.

To reject  $H_0$  means that at least one of the independent variables contributes significantly to the model. The null hypothesis can be tested by an analysis of variance (ANOVA) [10].

- Tests for each Coefficient

The hypothesis to test the significance of the coefficient  $\hat{\beta}_j$  are

$$H_0 : \hat{\beta}_j = 0$$

$$H_1 : \hat{\beta}_j \neq 0, j = 0, \dots, k.$$

If  $H_0$  is not rejected, one has the indication that the variable  $x_j$  does not need to be included in the model, because the statistics shows that this variable is little significant. The null hypothesis  $H_0$  is rejected if  $|t_0| > t_{(n-k-1)}(1 - \alpha/2)$ , where  $n$  is the sample size and  $\alpha$  is the test significance level. Significant values for  $t_0$  must be bigger than 2.

In the next section, the use of linear regression models to locate promising regions is presented.

### 3 Search-Space Reduction

In this work, we present a different approach to find promising regions. Instead of elaborating complex probabilistic models to determine promising regions in the search-space, we apply a Multiple Linear Regression to, starting from the search-space limits, iteratively eliminate portions of the search-space where the algorithm guarantees, with a high probability, that there is no global optimum. The algorithm ends when it cannot guarantee a further reduction. Our approach, a SRA, is a technique to determine one or more promising regions before the use of an optimization algorithm. This way, one can, for instance, concentrate the initial population into the limits defined by the SRA, where there is a high probability to find the global optimum.

- Domain Optimization Algorithm

DOAs have as objective to aid the global optimization algorithms indicating the initial search-space areas (domain) with larger success possibility of finding the global optimum, excluding areas considered unfavorable. This way, DOAs try to increase the efficiency of global optimization algorithms, and can be essential to obtain satisfactory results in situations where the execution of a great number of experiments is unviable.

Many techniques that use stochastic knowledge try guide the search by using highly complex models of the search-space [712]. The approach proposed in this work uses

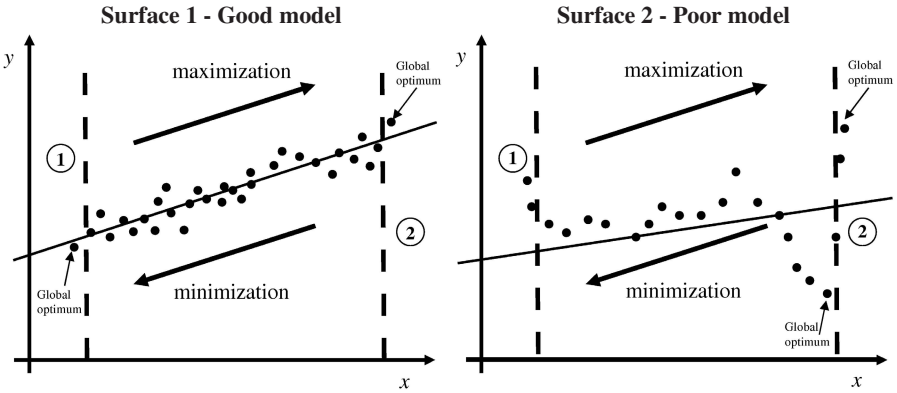


Fig. 1. Adjusting a linear model

the opposite strategy, where a simple model (linear) is employed. If that model presents an appropriate fitting, one can take the decision of eliminating the area which appears to be farther from the global optimum. Two examples can be seen in Figure 1.

Analyzing the sampled points, it is possible to notice that, in spite of being noisy, the Surface 1 can be appropriately explained by a linear model. That means that the values predicted by the model, represented by the straight line, are not very different from the sampled values (points), resulting in a low prediction error. This way, the decision of eliminating an area can be taken with a considerable confidence. In case it is a minimization problem, the cut (elimination of an area of the search-space) is accomplished in the superior limit (area 2), because the global optimum tends to be on the left side. Otherwise, the inferior limit is cut (area 1). The *slice size* can be defined by the user or estimated from a small sampling in the area to be eliminated (area 1 or 2) to give larger safety in the cut.

On the other hand, the Surface 2 could not be well explained. A cut to minimize (area 2) unfortunately could lose the optimum. The same would not happen if one wanted to cut area 1 to maximize the function. The basic algorithm works basically as presented in Algorithm 1.

The *limits* are represented by a vector  $L[ll_1, ul_1, \dots, ll_k, ul_k]$ , where *ll* is the lower limit, *ul* is the upper limit, and *k* is the number of dimensions of the problem. *Max regions* is an integer defining the number of promising regions to find. *Max reduction* is a real number defining the minimum size of the promising region, for instance,  $MRD=0.1$  means that the delta between the inferior and superior limits is 10% of the original size. *Max slice size* is a float defining the maximum percentage of the search-space to cut. *Max errors* is an integer which defines the stop criterion. The number of errors is incremented, for now, when: 1) after 10 trials no good model is found or 2) a further reduction loses the best point found. The stop criterion is under study.

A good model can be chosen by its *p*-value or  $R^2$  value (the fraction of variance explained by the model). In this work, one consider a good model if its *p*-value is under 0.05 or if its  $R^2$  value is above 0.9. By using a linear model in the noisy functions studied in this work, one cannot expect a really good explanation of the variance, so the  $R^2$  parameter is almost never used.

---

**Algorithm 1.** Pseudocode for DOA

---

**Input:** limits L, sample size Size, max regions MR, max reduction MRD, max slice size MSS, max errors ME

**Reduce L:**

**Do**

**Do**

*Do - Find a good model for the search-space*

Generate a random uniform sample X of size Size inside limits L

Evaluate each point of the sample, generating a vector Y

Fit a linear model

**Loop until** ( $p$ -value > 0.05 or  $R^2 < 0.9$ ) and (tries < 10)

Select the most significant variables using the test for each coefficient

**For each** significant variable  $i$

Select the slice size proportionally to  $\hat{\beta}_i$

**If**  $\hat{\beta}_i$  sign is negative **Then**

Lower the upper limit (*slice size*)

**Else**

Raise the lower limit (*slice size*)

**End If**

**Loop**

**Loop until** errors < ME

**Increment** regions

**Loop until** regions < MR

**Output:** best point found BP, new limits L

---

By reducing the initial search-space, the initialization of the points can be concentrated in the promising region found. Thus, it tries to avoid the creation of points distant from the global optimum and to guarantee a good sampling near the optimum. The population is initialized, for example, in the vertexes and in the central point of the promising region (and around these points) or anywhere inside the promising region.

The *slice size* in the basic algorithm defines the DOA's flexibility. A small size means that the algorithm will make small reductions and can stop prematurely. Bigger sizes can make fast and extreme reductions and, sometimes, get very close the global optimum. However, DOA can lose it too. To deal with this problem, a portion of the population can be randomly initialized outside the promising region, to try to put at least one individual near to the global optimum when the optimization loses it.

After the domain optimization, a local search can be started with an heuristic initialization (inside the promising region). To evaluate DOA's efficiency, we tested some benchmark functions and the results are presented in the next section.

## 4 Experiments

In this work, we present preliminary results about the reduction of the search-space and its effects in global optimization problems using three well known global optimization algorithms. With these experiments, we do not compare the optimization algorithms, but the how the reduction of the search-space to promising areas affects the search.



The experiments were performed using an Acer notebook with Intel Celeron M 420, 1 Gb of RAM, Ubuntu Linux 6.06 kernel 2.6.15.27, and GCC 4.0.3. All optimization algorithms were developed in C/C++. The GA uses GALib 2.4.6 [18]. The multiple linear regression and the tests of significance were implemented in Ox 4.04 [3].

The benchmark functions used in the tests with the DOA are the functions from 1 to 12, which are part of a set benchmark functions recently proposed to evaluate new global optimization algorithms (see [17]), as showed in Table 1.

**Table 1.** Benchmark Functions

<b>Unimodal</b>	F1: Shifted Sphere Function; F2: Shifted Schwefel's Problem 1.2; F3: Shifted Rotated High Conditioned Elliptic Function; F4: Shifted Schwefel's Problem 1.2 with Noise in Fitness; F5: Schwefel's Problem 2.6 with Global Optimum on Bounds
<b>Multimodal</b>	F6: Shifted Rosenbrock's Function; F7: Shifted Rotated Griewank's Function without Bounds; F8: Shifted Rotated Ackley's Function with Global Optimum on Bounds; F9: Shifted Rastrigin's Function; F10: Shifted Rotated Rastrigin's Function; F11: Shifted Rotated Weierstrass Function; F12: Schwefel's Problem 2.13

Three well know global optimization algorithms are tested in this work: the GA, the Particle Swarm Optimization (PSO) [4] and the Differential Evolution (DE) [16]. The configuration of the algorithms were determined after experimental analysis (Table 2).

**Table 2.** Configuration of the Algorithms for  $k(\text{dim})=10$

DOA	GA	PSO	DE
Sample size: 60 points	Population: 60	Population: 10	Population: 80
MR: 5	BLX Crossover: 0.6	Weight: 0.7	F: 0.9
MRD: 0.2	Mutation: 0.1	Mode: Asynchronous	CR: 0.6
MSS: 0.5	Tournament selection		
ME: 30			

There are two methods to experiment: RAND (simple GA, PSO or DE with random initialization) and DOA (simple GA, PSO or DE executed after the search-space reduction). Each one of the methods is executed 30 times, for each function from  $f1$  to  $f12$ , with a maximum amount of evaluations equal to 100,000, including the evaluations made by the DOA, and the fitness of the best individual is stored, resulting in a vector  $f^*$  of 30 dimensions. Thus, one has  $12 f_{RAND}^*$  and  $12 f_{DOA}^*$ . In this experiments, the population is completely initialized inside the promising region, but one can generate some points anywhere in the search-space. The population is evaluated and the worst individual is replace dwith the best point found by the DOA.

Given the high magnitude of some standard deviations obtained in this experiment, the simple comparison between the means can lead to low confidence conclusions. For that reason, two statistical tests were adopted to compare the methods. First, a test of variance ( $F$ -test) with significance level  $\alpha = 0.05$  is applied. If the variances are very different, the non-parametric mean test (Wilcoxon Rank Sum test) can be ignored, due to a large confidence interval obtained by a big standard deviation in the mean test. In these cases, we simply get the smaller mean. On the other hand, if the test of variance

**Table 3.** GA with random and DOA initialization. *GO* is the global optimum value. Success is how many times (%) the GO was found, with a tolerance error of  $1e-6$ .  $k(\text{dim})=10$ .

	Method	GO	Mean	$\sigma$	F	W	%Success
<i>f1</i>	RAND	-450	-4.499352E+02	4.255330E-02	7.796294E-01	7.9010E-01	0.00
	DOA		-4.499382E+02	4.038020E-02	7.796294E-01	7.9010E-01	0.00
<i>f2</i>	RAND	-450	-4.441785E+02	4.288765E+00	8.626160E-01	5.3250E-01	0.00
	DOA		-4.446772E+02	4.151975E+00	8.626160E-01	5.3250E-01	0.00
<i>f3</i>	RAND	-450	7.936512E+05	6.201617E+05	9.452610E-01	2.7381E-02	0.00
	DOA		<b>5.391400E+05</b>	6.281887E+05	9.452610E-01	<b>2.7381E-02</b>	0.00
<i>f4</i>	RAND	-450	-4.375950E+02	9.946715E+00	3.655150E-01	4.8530E-01	0.00
	DOA		-4.351940E+02	1.178864E+01	3.655150E-01	4.8530E-01	0.00
<i>f5</i>	RAND	-310	4.225328E+01	2.105014E+02	7.796640E-01	2.5393E-01	0.00
	DOA		9.205122E+01	2.218283E+02	7.796640E-01	2.5393E-01	0.00
<i>f6</i>	RAND	390	5.631587E+02	1.584207E+02	4.171780E-04	1.0000E+0	0.00
	DOA		<b>5.365228E+02</b>	7.985351E+01	<b>4.171780E-04</b>	1.0000E+0	0.00
<i>f7</i>	RAND	-180	-1.790605E+02	3.943740E-01	6.076220E-01	6.2284E-01	0.00
	DOA		-1.790324E+02	3.581850E-01	6.076220E-01	6.2284E-01	0.00
<i>f8</i>	RAND	-140	-1.196933E+02	8.735110E-02	2.586150E-01	7.2271E-01	0.00
	DOA		-1.196997E+02	1.080190E-01	2.586150E-01	7.2271E-01	0.00
<i>f9</i>	RAND	-330	-3.299832E+02	1.320500E-02	7.883134E-01	2.5422E-01	0.00
	DOA		-3.299800E+02	1.388620E-02	7.883134E-01	2.5422E-01	0.00
<i>f10</i>	RAND	-330	-3.091110E+02	8.713330E+00	3.315700E-01	9.2402E-01	0.00
	DOA		-3.097330E+02	7.260829E+00	3.315700E-01	9.2402E-01	0.00
<i>f11</i>	RAND	90	9.646602E+01	1.879740E+00	1.285280E-01	8.8914E-01	0.00
	DOA		9.650208E+01	1.411336E+00	1.285280E-01	8.8914E-01	0.00
<i>f12</i>	RAND	-460	4.072837E+02	1.219768E+03	6.465330E-02	9.0074E-01	0.00
	DOA		3.705809E+02	8.599837E+02	6.465330E-02	9.0074E-01	0.00

shows that the variances are equal, the wilcoxon-test is applied, also with a significance level  $\alpha = 0.05$ . If the means are considered different, according to the  $p$ -value returned by the wilcoxon-test, the best algorithm is the one which presents the smallest mean. To be considered different, the  $p$ -value must be lower than  $\alpha$  value.

The results of the experiments with stop criterion equal to 10 errors (Table 2) are presented in the Tables 3, 4, and 5. The *Mean* and  $\sigma$  (standard deviation) columns are calculated from the  $f^*$  vectors. With  $f_{RAND}^*$  and  $f_{DOA}^*$  of the corresponding function,  $F$  is the  $p$ -value returned by the  $F$ -test and  $W$  is the  $p$ -value returned by the wilcoxon-test. Finally, the best mean and the test used to decide it (F or W) are bolded, and when the means are considered equal, they are unbolded.

In the experiment with the GA (Table 3), one can see in the *Mean* column that the RAND algorithm achieved better results in the functions 4, 5, 7, 9, and 11. In the other seven functions, the GA was benefited by the DOA. Nevertheless, a statistical analysis presents that almost all differences in the means are not significant. This way, the  $p$ -values obtained by the F and wilcoxon-tests present that the only differences are in the functions 3 and 6.

In the majority of the tests with the GA, the DOA was not helpful. However, in the functions where the DOA initialization led to a smaller mean, even if there is no

**Table 4.** PSO with random and DOA initialization. *GO* is the global optimum value. Success is how many times (%) the GO was found.  $k(\text{dim})=10$ .

	Method	GO	Mean	$\sigma$	F	W	%Success
<i>f1</i>	RAND	-450	<b>-4.500000E+02</b>	0.000000E+00	<b>0.000000E+00</b>	3.3371E-01	100,00
	DOA		-4.500000E+02	5.659800E-06	0.000000E+00	3.3371E-01	96,67
<i>f2</i>	RAND	-450	-4.499998E+02	2.500630E-04	1.241000E-01	5.5711E-01	3,33
	DOA		-4.499998E+02	3.341940E-04	1.241000E-01	5.5711E-01	3,33
<i>f3</i>	RAND	-450	1.319745E+05	1.094572E+05	3.949000E-01	2.4180E-01	0.00
	DOA		1.050698E+05	9.330262E+04	3.949000E-01	2.4180E-01	0.00
<i>f4</i>	RAND	-450	<b>-4.499964E+02</b>	3.215453E-03	<b>1.287860E-07</b>	1.2777E-01	0.00
	DOA		-4.499922E+02	9.378405E-03	1.287860E-07	1.2777E-01	0.00
<i>f5</i>	RAND	-310	-3.100000E+02	0.000000E+00	1.000000E+00	1.0000E+00	0.00
	DOA		-3.100000E+02	0.000000E+00	1.000000E+00	1.0000E+00	0.00
<i>f6</i>	RAND	390	4.196969E+02	1.145642E+02	0.000000E+00	1.1243E-01	0.00
	DOA		<b>4.049251E+02</b>	1.595585E+01	<b>0.000000E+00</b>	1.1243E-01	0.00
<i>f7</i>	RAND	-180	-1.796958E+02	2.415142E-01	8.541380E-02	3.3540E-01	0.00
	DOA		-1.796811E+02	1.745095E-01	8.541380E-02	3.3540E-01	0.00
<i>f8</i>	RAND	-140	-1.196766E+02	6.735874E-02	6.153058E-01	8.2025E-04	0.00
	DOA		<b>-1.197422E+02</b>	7.401175E-02	6.153058E-01	<b>8.2025E-04</b>	0.00
<i>f9</i>	RAND	-330	-3.282420E+02	1.188072E+00	1.803260E-01	1.3526E-01	6,67
	DOA		-3.286402E+02	9.231950E-01	1.803260E-01	1.3526E-01	6,67
<i>f10</i>	RAND	-330	-2.845969E+02	2.082848E+01	1.037290E-05	2.4245E-05	0.00
	DOA		<b>-3.047281E+02</b>	8.699750E+00	<b>1.037290E-05</b>	2.4245E-05	0.00
<i>f11</i>	RAND	90	9.419266E+01	1.483934E+00	8.473570E-01	9.8244E-01	0.00
	DOA		9.413380E+01	1.538418E+00	8.473570E-01	9.8244E-01	0.00
<i>f12</i>	RAND	-460	2.876802E+03	4.704678E+03	8.881780E-016	3.8526E-02	0.00
	DOA		<b>-1.001094E+01</b>	7.780136E+02	<b>8.881780E-016</b>	3.8526E-02	0.00

statistical differences, the mean differences seems very clear. Thus, in these functions, the DOA can be very useful. Other interesting point is the success rate column. The GA seems to be capable of finding the correct region and get close to the global optimum, but has no success in finding it.

The PSO algorithm has a similar behavior. Table 4 shows the results. Analyzing the means, the RAND algorithm achieved better results in five functions: 1, 2, 4, 7, and 11. The PSO with DOA had better results in the other six functions: 3, 6, 8, 9, 10, and 12. Differently from the GA experiment, there are differences in the means in the functions 1, 4, 6, 8, 10, and 12. However, in this experiment, the statistical analysis shows that the RAND algorithm was the best only in the functions 1 (due to F) and 4. On the other hand, the PSO with DOA achieved better means in four functions (6, 7, 10, and 12). In the other four functions, the means were considered equal. It is important to notice that the PSO reached the global optimum in three functions. In function 1, the PSO with DOA initialization led to a lower success rate in comparison with the PSO with random initialization.

Table 5 presents the results of the DE. This experiment reflects the benefits of using the DE with DOA. In the results, one can see by the mean values, that the DE gets trapped in local optima. The difference in the means are very clear in the majority of

**Table 5.** DE with random and DOA initialization. *GO* is the global optimum value. Success is how many times (%) the GO was found.  $k(\text{dim})=10$ .

	Method	GO	Mean	$\sigma$	F	W	%Success
<i>f1</i>	RAND	-450	-3.653906E+02	3.255174E+02	0.000000E+00	7.8282E-02	83,33
	DOA		<b>-4.500000E+02</b>	1.825740E-07	<b>0.000000E+00</b>	7.8282E-02	96,67
<i>f2</i>	RAND	-450	-3.206112E+02	5.021913E+02	0.000000E+00	1.0121E-01	80,00
	DOA		<b>-4.481400E+02</b>	1.018766E+01	<b>0.000000E+00</b>	1.0121E-01	93,33
<i>f3</i>	RAND	-450	1.075683E+05	4.969392E+05	0.000000E+00	1.5151E-01	0.00
	DOA		<b>-4.49992E+02</b>	1.415220E-03	<b>0.000000E+00</b>	1.5151E-01	0.00
<i>f4</i>	RAND	-450	-3.655265E+02	3.238781E+02	0.000000E+00	8.1238E-06	73,33
	DOA		<b>-4.481195E+02</b>	1.030014E+01	<b>0.000000E+00</b>	8.1238E-06	6,67
<i>f5</i>	RAND	-310	2.400386E+02	1.293722E+03	1.708130E-01	2.4846E-01	83,33
	DOA		-4.733649E+01	9.995980E+02	1.708130E-01	2.4846E-01	93,33
<i>f6</i>	RAND	390	2.627200E+07	7.922668E+07	0.000000E+00	3.8709E-01	0.00
	DOA		<b>3.917685E+02</b>	3.159689E+00	<b>0.000000E+00</b>	3.8709E-01	0.00
<i>f7</i>	RAND	-180	-1.697347E+02	3.698725E+01	0.000000E+00	7.8601E-01	0.00
	DOA		<b>-1.793692E+02</b>	9.786190E-02	<b>0.000000E+00</b>	7.8601E-01	0.00
<i>f8</i>	RAND	-140	-1.196546E+02	7.576253E-02	6.344733E-01	2.7381E-03	0.00
	DOA		<b>-1.197189E+02</b>	8.282364E-02	6.344733E-01	<b>2.7381E-03</b>	0.00
<i>f9</i>	RAND	-330	-3.055507E+02	5.077101E+00	6.564068E-01	7.5927E-12	0.00
	DOA		<b>-3.172557E+02</b>	4.670964E+00	6.564068E-01	<b>7.5927E-12</b>	0.00
<i>f10</i>	RAND	-330	-3.040944E+02	4.378923E+00	3.466233E-01	8.5027E-02	0.00
	DOA		-3.059390E+02	3.669547E+00	3.466233E-01	8.5027E-02	0.00
<i>f11</i>	RAND	90	9.707722E+01	2.562276E+00	2.169988E-01	1.0920E-03	0.00
	DOA		<b>9.490644E+01</b>	2.031225E+00	2.169988E-01	<b>1.0920E-03</b>	0.00
<i>f12</i>	RAND	-460	1.798956E+03	4.973305E+03	1.110220E-015	8.4177E-01	0.00
	DOA		<b>-9.900257E+0</b>	8.309911E+02	<b>1.110220E-015</b>	8.4177E-01	0.00

the functions. Starting the population in a heuristic way (inside a promising region), reduce the chances of getting trapped. The DOA's initialization achieved better results in all functions. Statistically, the means are considered equal only in the functions 5, 8, and 10. There is a clear difference between the means in function 5, but the test of variance and the test of means return equality.

The success rate can be another indicative of the DOA's initialization effect. Excluding function 4, in all other functions in which the success rate is greater than zero, the DOA's initialization had better result. In function 4, the DE with DOA had a substantially better mean but a considerably lower success rate. In this case, the DE had difficult to reach the global optimum when the DOA was used, but stopped very close to it. The DE with random initialization got trapped in local minima in 5 executions. That is why it has a high success rate and also a high mean value.

In 36 tests (12 GA + 12 PSO + 12 DE) the results of the statistical test between the two initialization methods were: the random initialization outperformed the DOA initialization in 2 tests, the DOA initialization outperformed the random initialization in 16 tests, and in the other 18 tests the means were statistically equal.

## 5 Conclusions

In this work, we presented an approach, a Domain Optimization Algorithm which belongs to a class of algorithms called Search-space Reduction Algorithms, which is a technique to determine a promising region before the use of an optimization algorithm, instead of detecting one region during the global optimization process.

The proposed DOA uses Multiple Linear Regression to, starting from the search-space limits, iteratively eliminate portions of the search-space where the algorithm guarantees, with a high probability, that there is no global optimum. In general, algorithms like the EDAs try to guide the search by modeling the space of solutions through highly complex models. That is exactly the opposing of what we do in our algorithm.

As the proposed technique is a pre-optimization process, the objective of this paper is to show how a search-space reduction using simple linear models can increase the efficiency of global optimization algorithms. Analyzing the presented results, one can conclude that the proposed approach definitely shows relevant results for tests using hard benchmark functions.

For future work, we will explore other ways to achieve better reductions also keeping the global optimum inside the promising region. Other search-space analysis can be taken in consideration before eliminating a portion of the space, trying to guarantee even more that the global optimum remains inside the promising region. Also, we want to test our approach with more sophisticated optimization algorithms.

## Acknowledgment

The authors would like to acknowledge CAPES for the financial support given to this research.

## References

1. Chan, K.Y., Aydin, M.E., Fogarty, T.C.: A taguchi method-based crossover operator for the parametrical problems. In: Sarker, R., Reynolds, R., Abbass, H., Tan, K., McKay, B., Essam, D., Gedeon, T. (eds.) CEC 2003. Proceedings of the 2003 Congress on Evolutionary Computation, 8-12 December 2003, pp. 971–977. IEEE Computer Society Press, Los Alamitos (2003)
2. Chen, S., Smith, S.: Improving genetic algorithms by search space reduction (with applications to flow shop scheduling). In: GECCO 1999. Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann, San Francisco (1999)
3. Cribari-Neto, F., Zarkos, S.G.: Econometric and statistical computing using ox. *Comput. Econ.* 21(3), 277–295 (2003)
4. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
5. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1989)
6. Hinkelmann, K., Kempthorne, O.: Design and Analysis of Experiments: Introduction to Experimental Design. In: *Wiley Series in Probability and Mathematical Statistics*, vol. 1 (1994)

7. Liang, K.-H., Yao, X., Newton, C.: Evolutionary search of approximated n-dimensional landscapes. *International Journal of Knowledge-Based Intelligent Engineering Systems* 4(3), 172–183 (2000)
8. Lung, Y.W., Wang, Y.: An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Transactions on Evolutionary Computation* 5, 41–53 (2001)
9. Magalhaes, S.R.S.: A avaliação de métodos para a comparação de modelos de regressão por simulação de dados. Master's thesis (Mestrado em Estatística e Experimentação Agropecuária), Universidade Federal de Lavras, Lavras (2002)
10. Montgomery, D.C.: *Design and Analysis of Experiments*, vol. 1. John Wiley and Sons, Chichester (1997)
11. Myers, R.H., Montgomery, D.C.: *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, 2nd edn. Wiley, Chichester (2002)
12. Pelikan, M., Goldberg, D.E., Cant-paz, E.: Linkageproblem, distribution estimation and bayesian networks. *Evolutionary Computation* 3(8), 311–340 (2000)
13. Reeves, C.R., Wright, C.C.: Epistasis in genetic algorithms: an experimental design perspective. In: 6th International Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco (1995)
14. Reeves, C.R., Wright, C.C.: An experimental design perspective on genetic algorithms. In: Whitley, D., Vosa, M. (eds.) *Foundations of Genetic Algorithms 3*, Morgan Kaufmann, San Francisco (1995)
15. Srinivas, M., Patnaik, L.M.: Learning neural network weights using genetic algorithms- improving performance by search-space reduction. In: 1991 IEEE International Joint Conference on Neural Networks, November 18-21, 1991, pp. 2331–2336. IEEE Computer Society Press, Los Alamitos (1991) IEEE Cat. No. 91CH3065-0
16. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4), 341–359 (1997)
17. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.-P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. Technical Report KanGAL Report 2005005, Nanyang Technological University, Singapore (2005)
18. Wall, M.: Galib - a c++ library of genetic algorithm components (2006), <http://lancet.mit.edu/ga/>

# Clustering Search Approach for the Traveling Tournament Problem

Fabrício Lacerda Biajoli and Luiz Antonio Nogueira Lorena

Instituto Nacional de Pesquisas Espaciais - INPE  
Laboratório Associado de Computação e Matemática Aplicada - LAC  
Av. dos Astronautas, 1.758 - São José dos Campos-SP, Brasil

**Abstract.** The *Traveling Tournament Problem* (TTP) is an optimization problem that represents some types of sports timetabling, where the objective is to minimize the total distance traveled by the teams. This work proposes the use of a hybrid heuristic to solve the mirrored TTP (mTTP), called *Clustering Search* (\*CS), that consists in detecting supposed promising search areas based on clustering. The validation of the results will be done in benchmark problems available in literature and real benchmark problems, e.g. Brazilian Soccer Championship.

## 1 Introduction

Scheduling problems in sports has become an important class of optimization problems in recent years. For the applications of Operational Research the management of this sporting activity type is an area very promising and little explored. The professional sport leagues represent one of the largest economical activities around the world. For several sports, e.g. soccer, basketball, football, baseball, hockey, etc, where the teams plays a double round-robin tournament among themselves, where the games are played in different places during some time period, the automating of that schedulings are necessary and very important. Other facts fortify the application of optimization techniques. Teams and leagues do not want to waste their investments in players and structure in consequence of poor scheduling of games; sports leagues represent significant sources of revenue for radio and television world networks; the scheduling interfere directly in the performance of the teams; etc. On the other hand, sport leagues generate extremely challenging optimization problems, that attract attention of the Operational Research communities.

The scheduling problems in sports are known in the literature as Traveling Tournament Problem and it was proposed by Easton et al. [7]. The TTP abstracts the salient features of Major League Baseball (MLB) in the United States and was established to stimulate research in sport scheduling. Since the challenge instances were proposed the TTP has raised significant interest. Several works in different contexts (see e.g. [2, 3, 5, 9, 13, 16, 17, 18]) tackled the problem of tournament scheduling in different leagues and sports, which contains many interesting discussion on sport scheduling. Basically, the schedule of MLB is a

conflict between minimizing travel distances and feasibility constraints on the home/away patterns. A TTP solution is a double round-robin which satisfies sophisticated feasibility constraints (e.g. no more than three away games in a road trip) and minimizes the total travel distances of the teams.

Problems of that nature contain in general many conflicting restrictions to be satisfied and different objectives to accomplish, like minimize the total road trips of the teams during the tournament, one just game per day and per team, accomplishment of certain games in stadiums and in pre-established dates, number of consecutive games played in the team's city and out, etc. To generate good schedulings, satisfying all constraints, is a very hard task. The difficulty of solution of that problem is attributed to the great number of possibilities to be analyzed, e.g., for a competition with 20 teams there are  $2,9062 \times 10^{130}$  possible combinations [4].

This work proposes the application of Clustering Search (\*CS) [15] to solve the mirrored version of the TTP, known as Mirrored Traveling Tournament Problem (mTTP) [16]. The \*CS is a generalization of the Evolutionary Clustering Search (ECS) proposed in [14]. This paper is organized in six sections, being this the first. In the next section, the Traveling Tournament Problem and your *mirrored* version are described. In Section 3, we describe the basic ideas and conceptual components of \*CS. In Section 4, the methodology is detailed, with neighborhoods and the algorithm implemented. The computational results are examined in Section 5 and conclusions are summarized in Section 6.

## 2 Problem Description

The *Traveling Tournament Problem* was first proposed by Easton et al. in [7]. A scheduling to a *double round-robin* (DRR) tournament, played by  $n$  teams, where  $n$  is an even number, consists in a schedule where each team plays with each other twice, one game in its home and other in your opponent's home. A game between teams  $T_i$  and  $T_j$  is represented by unordered pair  $(i, j)$ . That schedule needs  $2(n - 1)$  rounds to represent all games of the tournament. The input data consists of the number of teams ( $n$ ), a symmetric matrix  $D$ ,  $n \times n$ , where  $D_{ij}$  represents the distance between the home cities of the teams  $T_i$  and  $T_j$ .

The cost of a team is the total distance traveled starting from its home city and return there after the tournament ending. The cost of the solution is the sum of the cost of every team.

The objective is to find a schedule with minimum cost, satisfying the following constraints:

- No more than three consecutive home or away games for any team;
- A game of  $T_i$  at  $T_j$ 's home cannot be followed by the game of  $T_j$  at  $T_i$ 's home;

The *Mirrored Traveling Tournament Problem* (mTTP) proposed by Ribeiro and Urrutia in [16] is a generalization of TTP that represents the common structure in Latin-America tournaments (e.g. Brazilian Soccer Championship). The main



difference is the concept of *mirrored double round-robin* (MDRR). A MDRR is a tournament where each team plays every other once in the  $n - 1$  rounds, followed by the same games with reversed venues in the last  $n - 1$  rounds.

The objective is the same of TTP, find a schedule with minimum cost satisfying the same constraints plus an additional constraint: the games played in round  $R$  are the same played in round  $R + (n - 1)$  for  $R = 1, 2, \dots, n - 1$ , with reversed venues.

### 3 Clustering Search

The Clustering Search (\*CS) [15] is a generalization of the Evolutionary Clustering Search (ECS) proposed in [14] that employs clustering for detecting promising areas of the search space. It is particularly interesting to find out such areas as soon as possible to change the search strategy over them. An area can be seen as a search subspace defined by a neighborhood relationship in metaheuristic coding space.

In the ECS, a clustering process is executed simultaneously to an evolutionary algorithm, identifying groups of individuals that deserve special interest. In the \*CS, the evolutionary algorithm was substituted by distinct metaheuristics.

The \*CS attempts to locate promising search areas by framing them by clusters. A cluster can be defined as a tuple  $G = \{c; r; s\}$  where  $c$ ,  $r$  and  $s$  are, respectively, the *center* and the *radius* of the area, and a *search strategy* associated to the cluster.

The center is a solution that represents the cluster, identifying the location of the cluster inside of the search space. Initially, the center  $c$  is obtained randomly and progressively it tends to slip along really promising points in the close subspace. The radius  $r$  establishes the maximum distance, starting from the center, that a solution can be associated to the cluster.

For example, in combinatorial optimization,  $r$  can be defined as the number of movements needed to change a solution into another. The search strategy is a systematic search intensification, in which solutions of a cluster interact among themselves along the clustering process, generating new solutions.

The \*CS consists of four conceptually independent components with different attributions: a search metaheuristics (SM); an iterative clustering (IC); an analyzer module (AM); and a local searcher (LS).

Figure 1 shows the four components and the \*CS conceptual design.

The SM component works as a full-time solution generator. The algorithm is executed independently of the remaining components and this must be able of the continuous generation of solutions directly for the clustering process. Simultaneously, clusters are maintained to represent these solutions. This entire process works like an infinite loop, in which solutions are generated along the iterations.

IC component aims to gather similar solutions into groups, maintaining a representative cluster center for them. To avoid extra computational effort, IC is designed as an online process, in which the clustering is progressively fed by

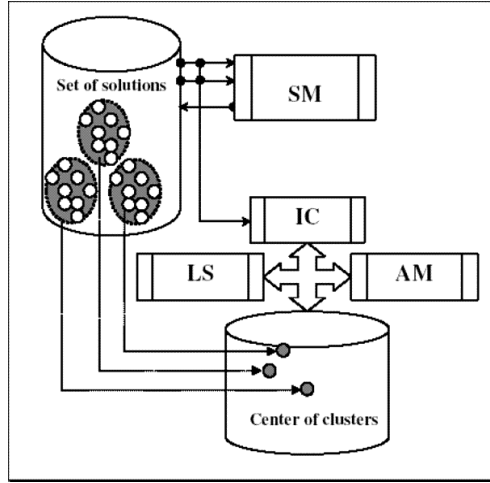


Fig. 1. \*CS Components

solutions generated in each iteration of SM. A maximum number of clusters  $NC$  is a bound value that prevents an unlimited cluster creation. A *distance metric* must be defined, a priori, allowing a similarity measure for the clustering process.

The AM component provides an analysis of each cluster, in regular intervals, indicating a probable promising cluster. A *cluster density*,  $\delta_i$ , is a measure that indicates the activity level inside the cluster  $i$ . For simplicity,  $\delta_i$  counts the number of solutions generated by SM and allocated to the cluster  $i$ . Whenever  $\delta_i$  reaches a certain *threshold*, meaning that some information template becomes predominantly generated by SM, such information cluster must be better investigated to accelerate the convergence process on it. AM is also responsible for the elimination of clusters with lower densities, allowing to create other centers and keeping framed the most active of them. The cluster elimination does not affect the set of solutions in SM. Only the center is considered irrelevant for the process.

At last, the LS component is a local search module that provides the exploitation of a supposed promising search area, framed by cluster. This process can happen after AM having discovered a promising cluster and the local search is applied on the center of the cluster. LS can be considered as the particular search strategy  $s$  associated with the cluster.

## 4 Methodology

The methodology used to solve the mTTP is based on the use of the Clustering Search (\*CS), explained in Section 3, with the metaheuristic Variable Neighborhood Search (VNS) [12] as the SM component of \*CS, generating the approach VNS-CS. The next sections describes in full detail the methodology.

## 4.1 Representation of a Solution

The representation of a schedule is a table indicating the opponents of the teams, where each line corresponds to a team and each column corresponds to a round. The opponent's representation is given by the pair  $(i, j)$ , where  $i$  represents the team  $T_i$  and  $j$  represents the round  $r_j$  (e.g., the opponent of the team  $T_1$  in round  $r_2$  is given by  $(1, 2)$ ). If  $(i, j)$  is positive, the game takes place at  $T_i$ 's home, otherwise at  $T_i$ 's opponent home.

In this work only the  $n - 1$  first rounds (*first half*) are represented, because the  $n - 1$  last rounds (*second half*) are the mirror with reversed venues and all alteration in the first half affects the second half (see Figure 2).

	First half					Second half				
$T_i/r_k$	1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-5	-3	6	-4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	-1	6	2	-5	3	1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

Fig. 2. Representation of a Schedule

## 4.2 The Neighborhood

Five different movements have been defined to compose distinct kinds of neighborhood, named *Home-away swap*, *Team swap*, *Round swap*, *Partial Round swap* and *Games swap*, from a schedule  $S$ . The neighborhood of a schedule  $s$  is the set of the schedules (feasibles and infeasibles) which can be obtained by applying one of these five types of movements.

**Home-away swap.** This move swaps the home/away roles of a game involving the teams  $T_i$  and  $T_j$ . The application of the move *Home-away swap* in a solution  $s$  obtain a solution  $s'$ , with a single game swapped, by reversing the game's place. In other words, if team  $T_i$  plays at home with  $T_j$  ( $T_j$  plays away) in  $s$ , then  $T_j$  plays at home and  $T_i$  plays away in  $s'$ .

**Team swap.** This move swaps the schedule of two teams,  $T_i$  and  $T_j$ . Only the games where  $T_i$  and  $T_j$  play against each other are not swapped. For example, if a team  $T_L$  was playing against team  $T_i$  at home in a round  $r_k$  of  $s$ , then in the neighbor solution  $s'$  it'll play against team  $T_j$  in the same round ( $r_k$ ) at home.

**Round Swap.** Given two rounds  $r_i$  and  $r_j$ , the application of the Round Swap in a solution  $s$  obtain a solution  $s'$  where all the opponents of the all teams were swapped in these two rounds. For example, if a team  $T_i$  was playing against the teams  $T_j$  and  $T_k$  in the rounds  $r_m$  and  $r_L$ , respectively, of  $s$ , then in the neighbor solution  $s'$  the team  $T_i$  plays against the teams  $T_j$  and  $T_k$  in the rounds  $r_L$  and  $r_m$ , respectively.

**Partial Round swap.** Consider the teams  $T_i, T_j, T_k$  and  $T_L$  and the rounds,  $r_m$  and  $r_z$ , such that games  $\{T_i, T_k\}$  and  $\{T_j, T_L\}$  take place in round  $r_m$  and games  $\{T_i, T_L\}$  and  $\{T_j, T_k\}$  take place in round  $r_z$ . The application of the move Partial Round Swap consists in swapping the rounds in which these games take place. The games  $\{T_i, T_k\}$  and  $\{T_j, T_L\}$  are set to round  $r_z$  and the games  $\{T_i, T_L\}$  and  $\{T_j, T_k\}$  are set to round  $r_m$ .

**Games swap.** This move consists in selecting an arbitrary game and enforcing it to be played in a round, followed by the necessary modifications to avoid teams playing more than one game in the same round. In consequence, more games would be swapped to maintain the feasible of the schedule. The modifications that have to be applied to the current schedule give rise to an ejection chain move. Ejection chains are based on the notion of generating compound sequences of moves by linked steps in which changes in selected elements cause other elements to be ejected from their current state, position, or value assignment [16].

	First half					Second half				
T <sub>i</sub> /r <sub>k</sub>	1	2	3	4	5	6	7	8	9	10
1	-6	4	2	-5	-3	6	-4	-2	5	3
2	-5	3	-1	-4	-6	5	-3	1	4	6
3	4	-2	-5	6	1	-4	2	5	-6	-1
4	-3	-1	6	2	-5	3	1	-6	-2	5
5	2	-6	3	1	4	-2	6	-3	-1	-4
6	1	5	-4	-3	2	-1	-5	4	3	-2

	First half					Second half				
T <sub>i</sub> /r <sub>k</sub>	1	2	3	4	5	6	7	8	9	10
1	-5	-3	2	4	-6	5	3	-2	-4	6
2	-6	4	-1	-5	3	6	-4	1	5	-3
3	-4	1	-5	6	-2	4	-1	5	-6	2
4	3	-2	6	-1	5	-3	2	-6	1	-5
5	1	-6	3	2	-4	-1	6	-3	-2	4
6	2	5	-4	-3	1	-2	-5	4	3	-1

Fig. 3. Schedule before (left) and after (right) the application of *Games swap*

The figure 3 shows the schedule produced by the application of *Game swap* move. Note that several games are swapped, when the team  $T_3$  was enforcing to play with team  $T_1$  in round  $r_2$ .

### 4.3 Clustering Search for mTTP

A \*CS metaheuristic is now described: VNS-CS (*Variable Neighborhood Search Clustering Search*). In this approach, the component SM is a hybrid metaheuristic that combines VNS and VND. As it is already known, VNS [12] starts from the initial solution and at each iteration, a random neighbor is selected in the  $N_{(k)}(s)$  neighborhood of the current solution. That neighbor is then submitted to some local search method. If the solution obtained is better than the current, update the current and continue the search of the first neighborhood structure. Otherwise, the search continues to the next neighborhood. The VNS stopped when the maximum number of iterations since the last improvement is satisfied.

The local optimum within a given neighborhood is not necessarily an optimum within other neighborhoods, and a change of neighborhoods can also be performed during the local search phase. This local search is then called Variable Neighborhood Descent (VND) [12].

In this work, we used the VNS with the three movements as follow, in this order: *Games swap*, *Round swap* and *Partial Round swap*. We used the VND as a local search method of the VNS, and it is composed by the other two movements proposed, in this order: *Team swap* and *Home-away swap*.

The IC is the CS's core, working as a classifier, keeping in the system only relevant information, and driving a search intensification in the promising search areas. Initially, a maximum number of clusters ( $NC$ ) is defined. In our case, we used  $NC = 20$ . Solutions generated by VNS are passed to IC that attempts to group as known information, according to *distance metric*. In this paper, the distance metric was the number of different games between the solution and the center of the cluster, not considering the home-away definition ( $A \times B$  equals  $B \times A$ ).

If the information is considered sufficiently new, it is kept as a center in a new cluster. Otherwise, redundant information activates the closest center  $c_i$  (center  $c$  that minimizes the distance metric), causing some kind of perturbation on it.

The information will be considered similar if there is at least 50% of coincidence between the games of the two compared solutions.

The perturbation means an assimilation process, in which the center of the cluster is update by the new generated solution. Here, we used the Path-Relinking method [8], that generates several points (solutions) taken in the path connecting the solution generated by VNS and the center of the cluster. Since each point is evaluated by the objective function, the assimilation process itself is an intensification mechanism inside the clusters. The new center  $c_i$  is the best evaluated solution sampled in the path.

The AM is executed whenever an solution is assigned to a cluster, verifying if the cluster can be considered promising. A cluster becomes promising when reaches a certain density  $\lambda_t$ ,

$$\lambda_t \geq PD \left[ \frac{NS}{NT_c} \right] \quad (1)$$

where,  $NS$  is the number of solutions generated in the interval of analysis of the clusters,  $NT_c$  is the number of cluster in the iteration  $t$ , and  $PD$  is the desirable cluster density beyond the normal density, obtained if  $NS$  was equally divided to all clusters. In this work, we used  $NS = 300$ ,  $PD = 2$  and  $NT_c = 20$ , chosen after several tests.

The component LS has been activated when the AM discover a promising cluster. The LS implemented was the Iterated Local Search (ILS) [11]. This metaheuristic starts from a locally optimal feasible solution. A random perturbation (movement *Game swap*) is applied to the current solution and followed by a local search similar to VNS, with the movements *Team swap*, *Partial Round swap* and *Home-away swap*. If the local optimum obtained after these steps satisfies some acceptance criterion, then it is accepted as the new current solution, otherwise the latter does not change. The best solution is eventually updated and the above steps are repeated for 1000 iterations.

Figure 4 shows the Pseudo Code of the algorithm implemented.

```

Procedure VNS-CS( $f(\cdot), N(\cdot), s_0$ )
1  $Iter \leftarrow 0$ ;           { Current iteration }
2  $C_{active} \leftarrow \emptyset$ ; { Define if a cluster is active }
3  $CP \leftarrow False$ ;      { Define if a cluster is promising }
4  $s \leftarrow s_0$ ;         { Current Solution }
5 while ( $Iter < IterMax$ ) do
6      $k \leftarrow 1$ ;
7     while ( $k \leq 3$ ) do
8         Generate any neighbor  $s' \in N_{(k)}(s)$ ;
9          $s'' \leftarrow VND(s')$ ;
10         $C_{active} \leftarrow \text{Component-IC}(s'')$ ;
11         $CP \leftarrow \text{Component-AM}(C_{active})$ ;
12        if( $CP = True$ ) then
13             $\text{Component-LS}(C_{active})$ ;
14        if ( $f(s'') < f(s)$ ) then
15             $s \leftarrow s''$ ;
16             $k \leftarrow 1$ ;
17        else
18             $k \leftarrow k + 1$ ;
19        end-if;
20    end-while;
21     $Iter \leftarrow Iter + 1$ ;
22 end-while;
end-VNS-CS;

```

Fig. 4. Algorithm *VNS-CS*

## 5 Experiments and Computational Results

The algorithm was coded in C++ and was run on *Pentium IV 3.0 GHz clock with 512 Mbytes of RAM memory*.

The benchmark's instances, described in [7] and adapted to the mirrored form in [16] was used to validate the results. A real-life instance (br2003.24), where 24 teams playing in the main division of the 2003 edition of Brazilian Soccer Championship was also tested. These test problems are available to download in <http://mat.gsia.cmu.edu/TOURN/>. The parameters of the methods were empirically chosen, after several simulations.

Table 1 shows the results for the considered instances. For each instance is reported the best solution found by the algorithms proposed by [16] and [10] (they obtained the best known solutions) and the minimum computation time of this last algorithm, the solutions obtained by this approach, the relative gap in percent between our solutions and the best solution between [16] and [10], and the last column presents the total computation times in seconds. The results represented by “-” wasn't considered by the authors.

**Table 1.** Computational results

Instances	Best by [16]	Best by [10]	Time [10]	VNS-CS	gap(%)	Time
circ4	20	–	–	20	0%	6
circ6	72	–	–	72	0%	9
circ8	140	140	0.2	140	0%	438
circ10	272	272	28160	276	1,47%	4951
circ12	456	432	93.1	446	3,14%	1275
circ14	714	696	53053.5	702	0,86%	3672
circ16	978	968	38982.7	978	1,02%	826
circ18	1306	1352	178997.5	1352	3,40%	1153
circ20	1882	1852	59097.9	1882	1,59%	1189
nl4	8276	–	–	8276	0%	5
nl6	26588	–	–	26588	0%	7
nl8	41928	41928	0.1	41928	0%	1030
nl10	63832	63832	477.2	65193	2,09%	228
nl12	120655	119608	15428.1	120906	1,07%	2630
nl14	208086	199363	34152.3	208824	4,53%	796
nl16	279618	279077	55640.8	287130	2,80%	3201
con4	17	–	–	17	0%	4
con6	48	–	–	48	0%	5
con8	80	80	0.1	81	1,23%	39
con10	130	130	0.1	130	0%	92
con12	192	192	0.3	193	0,52%	299
con14	253	253	6.0	255	1,18%	131
con16	342	342	2.7	343	0,29%	407
con18	432	432	8.1	433	0,23%	1265
con20	524	522	1106.3	525	0,57%	1102
br2003.24	503158	–	–	512545	1,83%	798

The results presented demonstrate that the proposed methodology can be competitive, because was possible to obtain values near of the best results known in the literature, getting to reduce to zero the gap in six of the seventeen instances and being near of reducing to zero in some others. The worse gap was 4,53%.

The main aspect of the results is the computation time. See that the VNS-CS can reach solutions with similar quality quicker than the algorithm proposed in [10], e.g., whilst the algorithm of the literature took some days (more than two days) to reach the best solutions for the instance circ18, the VNS-CS reached high-quality solutions, near the best known, in few minutes.

The result of the real-life instance was very interesting. First, because it is the larger instance in the literature, with 24 teams. Second, because was observed a reduction of 51.09% in the total distance traveled, where in the official schedule the teams traveled 1.048.134 km and in the schedule found by the work they traveled 512.545 km only.

## 6 Conclusions

In this work was investigate the Mirrored Traveling Tournament Problem, first published in [16], with a implementation of a new way of detecting promising search areas based on clustering: the Clustering Search (\*CS) [15]. Together with other search metaheuristics, working as full-time solution generators, \*CS attempts to locate promising search areas by solution clustering. The clusters work as sliding windows, framing the search areas and giving a reference point to problem-specific local search procedures, besides an iterative process, called assimilation.

A metaheuristic based on \*CS was proposed to solve the all mTTP instances available in literature: VNC-CS (*Variable Neighborhood Search Clustering Search*).

Five different neighborhood structures for local search was investigated: three simple neighborhood, *Home-away swap*, *Round swap* and *Team swap*; and two more complicated, *Partial Round swap* and *Game swap*.

The approach become very promising, when the reported results are analyzed. Seventeen benchmark instances was tested and VNS-CS approach have achieved similar and sometimes superior performance than the works presented in literature. One real-life instance, the 2003 edition of Brazilian Soccer Championship, was also tested, with reduction of 51.09% of the total distance traveled by the official schedule. Although not all real constraints were analyzed, but only the main constraints, the obtained result for this instance was very interesting.

Finally, this work explores the mirrored instances of TTP, because its represents common structure in Latin-America tournaments.

For further research there are a variety of open issues that need to be addressed, e.g., to study other neighborhood to obtain high-quality solutions. In the same way the real championships has many other restrictions that should be analyzed: treatment of the classic games, allocation of stadiums, do not consider distance metric, but the airfares, and many others real constraints.

## References

1. Anagnostopoulos, A., Michel, L., Van Hentenryck, P., Vergados, Y.: A Simulated Annealing Approach to the Traveling Tournament Problem. In: Proceedings of Cpaior 2003 (2003)
2. Biajoli, F.L., Chaves, A.A., Mine, O.M., Souza, M.J.F., Pontes, R.C., Lucena, A., Cabral, L.F.: Scheduling the Brazilian Soccer Championship: A Simulated Annealing Approach. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 433–437. Springer, Heidelberg (2005)
3. Biajoli, F.L., Lorena, L.A.N.: Mirrored Traveling Tournament Problem: An Evolutionary Approach. In: Sichman, J.S., Coelho, H., Rezende, S.O. (eds.) IBERAMIA 2006 and SBIA 2006. LNCS (LNAI), vol. 4140, pp. 208–217. Springer, Heidelberg (2006)
4. Conclio, R., Zuben, F.J.: Uma Abordagem Evolutiva para Geração Automática de Turnos Completos em Torneios. *Revista Controle e Automação* 13(2), 105–122 (2002)



5. Costa, D.: An Evolutionary Tabu Search Algorithm and the NHL Scheduling Problem. *Infor.* 33(3), 161–178 (1995)
6. Dinitz, J., Lamken, E., Wallis, W.D.: Scheduling a Tournament. In: Colbourn, C.J., Dinitz, J. (eds.) *Handbook of Combinatorial Designs*, pp. 578–584. CRC Press, Boca Raton, USA (1995)
7. Easton, K., Nemhauser, G., Trick, M.: The Traveling Tournament Problem Description and Benchmarks. In: *CP 1999*, pp. 580–589 (2001)
8. Glover, F.: Tabu search and adaptive memory programming: Advances, applications and challenges. In: *Interfaces in Computer Science and Operations Research*. [S.I.], p. 175. Kluwer, Dordrecht (1996)
9. Hamiez, J.P., Hao, J.K.: Solving The Sports League Scheduling Problem with Tabu Search. In: Nareyek, A. (ed.) *Local Search for Planning and Scheduling*. LNCS (LNAI), vol. 2148, pp. 24–36. Springer, Heidelberg (2001)
10. Henteryck, P.V., Vergados, Y.: Traveling tournament scheduling: A systematic evaluation of simulated annealing. In: Beck, J.C., Smith, B.M. (eds.) *CPAIOR 2006*. LNCS, vol. 3990, pp. 228–243. Springer, Heidelberg (2006)
11. Loureno, H.R., Martin, O., Sttzle, T.: Iterated local search. In: *Handbook of Metaheuristics*, pp. 321–353. Kluwer Academic Publishers, Norwell (2002)
12. Mladenovic, N., Hansen, P.: Variable neighborhood search. *Computers and Operations Research* 24, 1097–1100 (1997)
13. Nemhauser, G., Trick, M.: Scheduling a Major College Basketball Conference. *Operations Research* 46(1), 1–8 (1998)
14. Oliveira, A.C.M.: Algoritmos Evolutivos Hbridos com Deteco de Regies Promissoras em Espaos de Busca Contnuo e Discreto. 200 p. Tese (Doutorado) Instituto Nacional de Pesquisa Operacional (INPE), So Jos dos Campos - SP (2004)
15. Oliveira, A.C.M., Lorena, L.A.N.: Pattern sequencing problems by clustering search. *Lecture Notes in Artificial Intelligence Series*, vol. 4140, pp. 218–227. Springer, Heidelberg (2006)
16. Ribeiro, C.C., Urrutia, S.: Heuristics for the Mirrored Traveling Tournament Problem. In: Burke, E.K., Trick, M.A. (eds.) *PATAT 2004*. LNCS, vol. 3616, pp. 323–342. Springer, Heidelberg (2005)
17. Schönberger, J., Mattfeld, D.C., Kopfer, H.: Memetic Algorithm Timetabling for Non-Commercial Sport Leagues. *European Journal of Operational Research* 153(1), 102–116 (1989)
18. Zhang, H.: Generating College Conference Basketball Schedules by a Sat Solver. In: *Proceedings Of The Fifth International Symposium on the Theory and Applications of Satisfiability Testing*, Cincinnati, pp. 281–291 (2002)

# Stationary Fokker – Planck Learning for the Optimization of Parameters in Nonlinear Models

Dexmont Peña, Ricardo Sánchez, and Arturo Berrones

Posgrado en Ingeniería de Sistemas, Facultad de Ingeniería Mecánica y Eléctrica  
Universidad Autónoma de Nuevo León AP 126, Cd. Universitaria,  
San Nicolás de los Garza, NL 66450, México  
`arturo@yalma.fime.uanl.mx`

**Abstract.** A new stochastic procedure is applied to optimization problems that arise in the nonlinear modeling of data. The proposed technique is an implementation of a recently introduced algorithm for the construction of probability densities that are consistent with the asymptotic statistical properties of general stochastic search processes. The obtained densities can be used, for instance, to draw suitable starting points in nonlinear optimization algorithms. The proposed setup is tested on a benchmark global optimization example and in the weight optimization of an artificial neural network model. Two additional examples that illustrate aspects that are specific to data modeling are outlined.

**Keywords:** global optimization, stochastic search, nonlinear modeling of data, statistical physics.

## 1 Introduction

Nonlinear models are becoming increasingly important in all fields of science and engineering. For instance, nonlinear interactions are considered essential for the emergence of collective behavior in extended dynamical systems [13]. On the other hand, it's widely accepted [5] that the signals generated by many complex systems can be modeled by nonlinear evolution equations. Other important example is the general question of optimal design in engineered systems, where linear models are usually nothing but crude approximations [8]. A limitation to the use of nonlinear models is that in order to be quantitatively meaningful, the models should be fitted to data. This step involves the optimization of a certain cost or likelihood function, which is oftenly defined on a high dimensional parameter space.

In this contribution we present a simple implementation of a recently introduced stochastic technique [1]. We show that our procedure may substantially improve the performance of global optimization algorithms. Our algorithm is tested on optimization problems of the type that arise in nonlinear modeling of data.

The method that we propose is fundamented in the interplay between Langevin and Fokker – Planck frameworks for stochastic processes, which is well known

in the study of out of equilibrium physical systems [12,16]. Given a cost function  $V(x_1, x_2, \dots, x_n, \dots, x_N)$ , a general stochastic search can be modeled by the Langevin equation

$$\dot{x}_n = -\frac{\partial V}{\partial x_n} + \varepsilon(t) \quad (1)$$

where  $\varepsilon(t)$  is an additive noise with zero mean and second moment  $E[\varepsilon(t)\varepsilon(t')] = D\delta(t-t')$ . The quantity  $D$  is called the diffusion coefficient. With a constant  $D$ , the process (1) gives a search by pure diffusion. Considering a diffusion coefficient that is slowly varying in time, Eq. (1) represents a simulated annealing process [6,15]. Moreover, other search strategies (e.g. those which involve memory) may be modeled by Eq. (1) after the inclusion of correlations in the additive noise. Hereafter we will consider  $\varepsilon(t)$  as a Gaussian white noise with zero mean and second moment  $E[\varepsilon(t)\varepsilon(t')] = D\delta(t-t')$ , where  $D$  is constant and  $\delta(t-t')$  is the Dirac's delta. With these prescriptions, the probability density associated to the stochastic process (1) is governed by the Fokker – Planck equation [12,16]

$$\dot{p} = \frac{\partial}{\partial x} \left[ \frac{\partial V}{\partial x} p \right] + D \frac{\partial^2 p}{\partial x^2} \quad (2)$$

The fundamental principle of our approach is that for bounded search spaces with reflecting boundary conditions, the density  $p$  converge to a stationary state [3]. In this way, we can study the limit  $t \rightarrow \infty$  of the search process (1) by looking for stationary solutions of Eq. (2), given that the associated optimization problem is defined over a region

$$L_{1,n} \leq x_n \leq L_{2,n}. \quad (3)$$

We now proceed to sketch the generalities of the algorithm proposed in [1]. The one dimensional projection of Eq. (2) at  $t \rightarrow \infty$  is given by

$$D \frac{\partial p(x_n | \{x_{j \neq n} = x_j^*\})}{\partial x_n} + p(x_n | \{x_{j \neq n} = x_j^*\}) \frac{\partial V}{\partial x_n} = 0. \quad (4)$$

In the stationary state the one dimensional projection gives the dependent probability densities that correspond to the original  $N$  dimensional process [1]. From Eq. (4) follows a linear second order differential equation for the cumulative distribution of  $x_n$ , denoted by  $y(x_n)$ ,

$$\frac{d^2 y}{dx_n^2} + \frac{1}{D} \frac{\partial V}{\partial x_n} \frac{dy}{dx_n} = 0 \quad (5)$$

$$y(L_{1,n}) = 0, \quad y(L_{2,n}) = 1$$

Random deviates can be drawn from the density  $p(x_n | \{x_{j \neq n} = x_j^*\})$  by the fact that  $y$  is an uniformly distributed random variable in the interval  $y \in [0, 1]$ , which follows from the transformation law of probabilities [1]. Viewed as

a function of the random variable  $x_n$ ,  $y(x_n)$  can be approximated through a linear combination of functions from a complete set that satisfy the boundary conditions in the interval of interest,

$$\hat{y} = \sum_{l=1}^L a_l \varphi_l(x_n). \quad (6)$$

Choosing for instance, a basis in which  $\varphi_l(0) = 0$ , the  $L$  coefficients are uniquely defined by the evaluation of Eq. (5) in  $L - 1$  interior points. In this way, the approximation of  $y$  is performed by solving a set of  $L$  linear algebraic equations, involving  $L - 1$  evaluations of the derivative of  $V$ .

The proposed procedure is based on the iteration of the following steps:

- 1) Fix the variables  $x_{j \neq n} = x_j^*$  and approximate  $y(x_n)$  by the use of formulas (5) and (6).
- 2) By the use of  $\hat{y}(x_n)$  construct a lookup table in order to generate a deviate  $x_n^*$  drawn from the stationary distribution  $p(x_n | \{x_{j \neq n} = x_j^*\})$ .
- 3) Actualize  $x_n = x_n^*$  and repeat the procedure for a new variable  $x_{j \neq n}$ .

We call this basic procedure a Stationary Fokker – Planck Machine (SFPM), where the name indicates the way in which the equilibrium distribution of the stochastic search process given by Eq. (1) is learned through the iteration of the previous steps. The algorithm has its grounds in the stationary Fokker – Planck equation. This contrast to the Fokker – Planck Learning Machine [15], which make use of the time dependent version of the same equation. Moreover, our procedure does not deal with the search of global optima by finite populations of points, which is the case for the Fokker – Planck Learning Machine and other related methods [15,14], but estimates entire distributions.

The convergence of the SFPM to the equilibrium distribution follows from the fact that the procedure represents a Gibbs sampling. Under very general conditions [2], the points generated by a one dimensional Gibbs sampling are asymptotically distributed according to the associated  $N -$  dimensional distribution.

In this paper we give a simple implementation of the SFPM algorithm which is capable to generate good approximations of the equilibrium distribution in a computationally inexpensive manner. Our implementation is tested on examples that are relevant to the important problem of nonlinear modeling.

## 2 A Simple Implementation of Stationary Fokker – Planck Learning

According to Gibbs sampling, the iteration of the SFPM will asymptotically generate points drawn from the density

$$p(x_n) = \int_{\{x_{j \neq n}\}} p(x_1, x_2, \dots, x_n, \dots, x_N) dx_1 dx_2 \dots dx_N \quad (7)$$

In the SFPM setting, the conditional distribution is modeled through the linear combination given by Eq. (6). From the convergence stated by Eq. (7), we propose to calculate

$$\langle \hat{y} \rangle = \sum_{l=1}^L \langle a_l \rangle \varphi_l(x_n). \quad (8)$$

where the brackets represent the average over the iterations of the SFPM. In this way, the derivative of the averaged function  $\langle \hat{y} \rangle$  will give an approximation of the density  $p(x_n)$ , which is the equilibrium density for the variable  $x_n$  after an infinitely long exploration time of the stochastic search process. The obtained density can be used to draw suitable populations of points for the associated optimization task. A more formal discussion of these statements will be presented elsewhere. At this point, our goal is to show the potential benefits of our algorithm through several nontrivial examples.

The precise description of our algorithm is the following:

Procedure

Set average\_coefficients = 0

For i = 1 to M // M is the number of iterations of the algorithm.

Initialize x randomly

For j = 1 to N // N is the amount of variables of the problem

Fix  $x_n! = x_j$

Run SFPM on  $x_j$  and store the coefficients

End For

average\_coefficients += coefficients

End For

average\_coefficients =  $\frac{1}{M}$  average\_coefficients

End Procedure

## 3 Examples

### 3.1 A Benchmark Example: Levy No. 5 Function

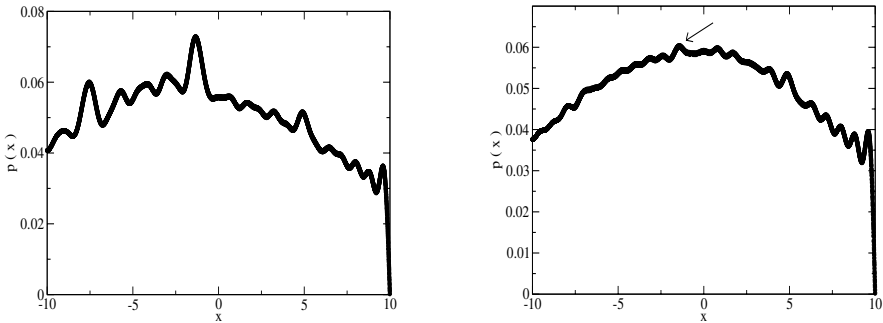
We begin our experimentation with a well known global optimization test problem, the Levy No. 5 function [9],

$$f(x) = \sum_{i=1}^5 i \cos((i-1)x_1 + i) \sum_{j=1}^5 j \cos((j+1)x_2 + j) \quad (9)$$

$$+ (x_1 + 1.42513)^2 + (x_2 + 0.80032)^2,$$

with a search space defined over the hypercube  $[-10, 10]$ . The direct implementation of a stochastic search through Eq. (1) would imply the simulation of a stochastic dynamical system composed by two particles with highly nonlinear interactions. By our methodology, in contrast, we are able to obtain adequate densities by linear operations and performing a moderate number of evaluations

of the cost function. In Fig. 1 the densities generated by 10 iterations of the SFPM with parameters  $L = 50$  and  $D = 200$  are shown. The obtained densities are perfectly consistent with the global properties of the problem, since the known global optimum at the point  $(-1.3068, -1.4248)$  is contained in the region of highest probability. The computational effort is low in the sense of the required number of cost function evaluations, given by  $2(L - 1)MN = 4900$ . This is comparable to the effort needed by advanced techniques based on populations in order to obtain good quality solutions for the same problem [9]. Our approach, however, is not limited to the convergence to good solutions, but it estimates entire densities. The implications of this in, for instance, the definition of probabilistic optimality criteria, are currently under research by us.



**Fig. 1.** Probability densities,  $p(x_1)$  and  $p(x_2)$  respectively, generated by 10 iterations of the SFPM for the Levy No. 5 function. The parameters of the SFPM are  $L = 50$  and  $D = 200$ . The global optimum is in the region of maximum probability.

In order to further verify the performance of the SFPM the following experiment has been done. The probability density function has been estimated by the SFPM with the parameter values  $L = 50, D = 200, M = 50$ . The point with maximum probability has been used as a starting point for a deterministic non-linear optimization algorithm. The results of this experiment are presented in Table 1. The deterministic routine that we have used is the Powell’s algorithm [11], a method based on conjugate directions that make no use of derivatives. The average success rate is 1.00, where each realization consists on 100 runs of the Powell’s algorithm. Using initial points drawn from uniform density gives as success rate of 0.043, which is reported in the Table 2. This implies a significant improvement by the use of the SFPM approach.

It is important to remark that the SFPM is a method to find a probability density function. With the knowledge of this function it’s possible to ensure by statistical methods that the global optimum resides in the maximum probability area. In Table 3 the computing effort for the solution by different evolutionary algorithms of the Levy No.5 problem is shown, measured in terms of the total number of cost function evaluations. By the SFPM approach, on the other hand,

**Table 1.** Success rate for the Levy No. 5 problem, using the point of maximum probability as a seed for the Powell’s algorithm. Each realization consist of 100 runs of the Powell’s routine. A solution is considered successful if its cost value differs in less than 0.0001 of the known global optimum.

Realization	1	2	3	4	5	6	7	8	9	10
Success rate	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

**Table 2.** Success rate for the Levy No. 5 problem, using uniform random numbers as initial points for the Powell’s algorithm

Realization	1	2	3	4	5	6	7	8	9	10
Success rate	0.00	0.04	0.01	0.05	0.02	0.03	0.07	0.08	0.08	0.05

we have found a density that used in connection with a simple deterministic optimization procedure gives a 100% success rate. The density has been learned performing 9800 cost function evaluations. It has also been observed that success rates greater than 80% can be achieved with 30 iterations of the SFPM which imply a total of 5880 cost function evaluations. These results indicate that a correct density for this example can be learned with a computational effort comparable (of the same order of magnitude) to the effort needed by evolutionary algorithms in order to find a global solution.

**Table 3.** Number of cost function evaluations and success rates for the Levy No. 5 problem reported for some evolutionary algorithms. The considered methods are Particle Swarm Optimization (PSO) [9], and two implementations of Differential Evolution (DE) [10]

Method	Cost function evaluations	Success
PSO	1049	93%
DE 1	2596	50%
DE 2	3072	100%
DE 3	3076	63%
DE 4	3568	16%
DE 5	3716	22%
DE 6	2884	100%

### 3.2 XOR Problem

The weight optimization of a neural network for the learning of the XOR table [4] is a classical example that shows the difficulties encountered in the fitting of nonlinear models. For an architecture described by two linear input nodes, two hidden nodes, one output node and logistic activation functions, the optimization problem reads

$$\min f, \tag{10}$$

$$\begin{aligned}
 f = & \left\{ 1 + \exp \left( -\frac{x_7}{1 + \exp(-x_1 - x_2 - x_5)} - \frac{x_8}{1 + \exp(-x_3 - x_4 - x_6)} - x_9 \right) \right\}^{-2} \\
 & + \left\{ 1 + \exp \left( -\frac{x_7}{1 + \exp(-x_5)} - \frac{x_8}{1 + \exp(-x_6)} - x_9 \right) \right\}^{-2} \\
 & + \left\{ 1 - \left[ 1 + \exp \left( -\frac{x_7}{1 + \exp(-x_1 - x_5)} - \frac{x_8}{1 + \exp(-x_3 - x_6)} - x_9 \right) \right]^{-1} \right\}^2 \\
 & + \left\{ 1 - \left[ 1 + \exp \left( -\frac{x_7}{1 + \exp(-x_2 - x_5)} - \frac{x_8}{1 + \exp(-x_4 - x_6)} - x_9 \right) \right]^{-1} \right\}^2
 \end{aligned}$$

The following experiment has been done: by the use of the SFPM a suitable random number generator has been constructed, from which 100 seeds are used as initial points for the Powell’s algorithm. The distribution of the random numbers has been constructed by 10 iterations of the SFPM with the parameters  $L = 60, D = 0.03$ . The search space is defined by the hypercube  $[-10, 10]$ . Ten independent realizations of the experiment are summarized on Table 4.

**Table 4.** Success rate for the XOR problem, drawing the initial points for the Powell’s algorithm from a density generated by the SFPM

Realization	1	2	3	4	5	6	7	8	9	10
Success rate	0.79	0.48	0.27	0.86	0.7	1.00	0.83	0.36	0.75	0.72

From the above table follows an average success rate of 0.675.

On another experiment, the 100 initials points have been drawn from a uniform distribution over the same search space. The results are summarized in the Table 5.

**Table 5.** Success rate for the XOR problem using uniform random numbers as initial conditions for the Powell’s routine

Realization	1	2	3	4	5	6	7	8	9	10
Success rate	0.27	0.21	0.18	0.27	0.18	0.15	0.24	0.33	0.19	0.26

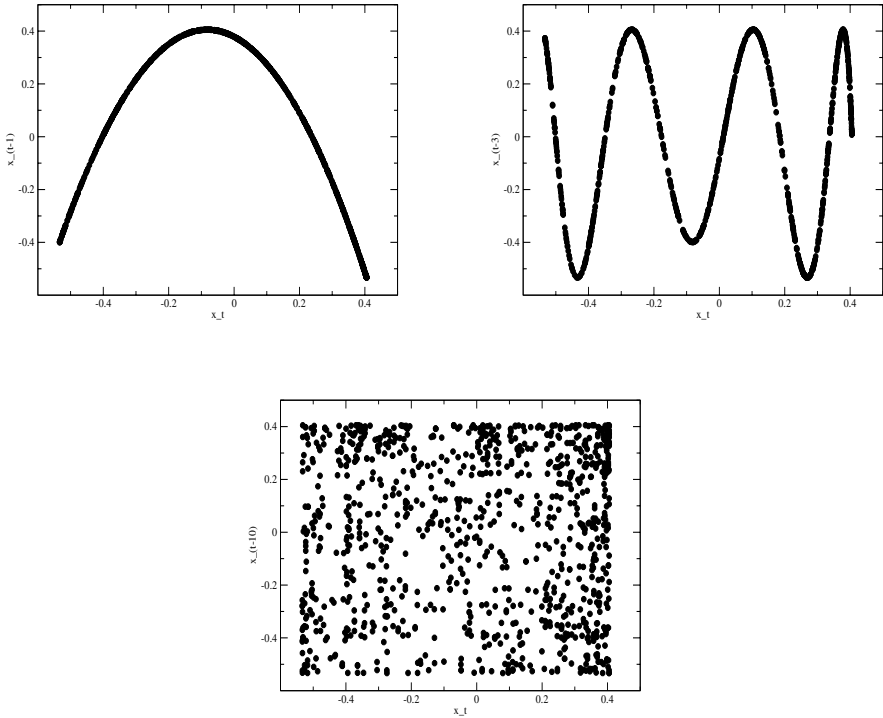
The average success rate in this case is 0.228. There is a substantial improvement on the performance of the Powell’s algorithm when using starting points generated by the estimated distribution. This result strongly suggest that the SFPM has correctly learned the density associated with the XOR problem.



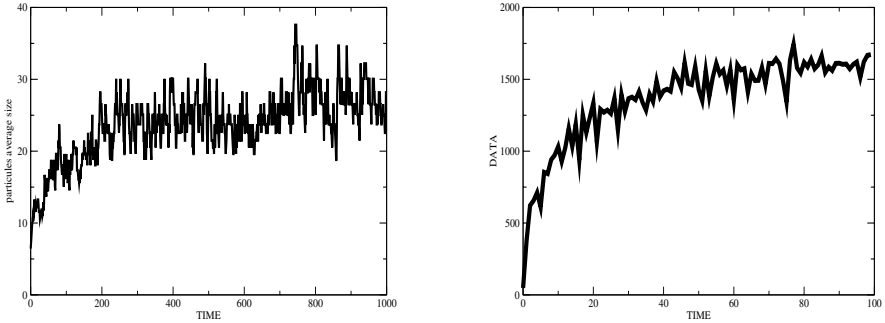
### 3.3 Additional Examples: Nonlinear Time Series Analysis and Maximum Likelihood Estimation

In this subsection we describe two examples currently under our research. In spite that these examples are not yet fully developed, we believe is worth to mention them because their description illustrate important aspects of the field of data analysis for which the application of the SFPM may be fruitful.

**Nonlinear Time Series Analysis.** The logistic map is a polynomial mapping, often cited as an archetypal example of how complex, chaotic behaviour can arise from very simple nonlinear dynamical equations [7]. Mathematically this can be written as  $x_{n+1} = rx_n(1 - x_n)$  where  $x_n$  is a number between zero and one and  $r$  is a positive number. We propose to study the optimization of the weights of a neural network that learns the logistic map using the SFPM. Because of its linear nature, some of the difficulties found in the characterization of nonlinear time series are expected to be tackled by the SFPM. For instance, as the sample grows, the number of nonlinear terms in the mean squared error cost function also grows. On the other hand, an incorrect selection for the number of input



**Fig. 2.** Plot of  $x_t$  vs.  $x_{t-1}$ ,  $x_t$  vs.  $x_{t-3}$  and  $x_t$  vs.  $x_{t-10}$  respectively. The parameter  $r$  of the logistic map is taken as 3.6.



**Fig. 3.** 3a) Average size of accumulated particles in the simulation of a reactor. 3b) Realization of a nonstationary Gaussian model for the observed data. Arbitrary scales.

units is critical for the nonlinearity of the map that is intended to be learned by the neural network. This aspect is illustrated on Figure 2.

We intend to exploit the linearity of the SFPM operations in order to improve the learning of ANN's for nonlinear time series, taking as a first example the archetypal logistic map.

**Maximum Likelihood Estimation.** This is an important tool that allows to determine unknown parameters of a distribution based on known outcomes. The calculation of Maximum Likelihood estimators yield to a highly nonlinear optimization problem. This is particularly true for nonstationary data. Nonstationary regimes are oftenly very important in applications. In Fig. 3a we show the simulation of a reactor used for the removal of pollutant metals from water, a research problem in which we are currently involved. The plotted quantity represents the evolution of the mean size of the metal aggregates. As can be seen, the data is strongly nonstationary. We intend to capture the basic features of the data through simple probabilistic models. In Fig. 3b we show a sample from a nonstationary Gaussian distribution with parameters of the type

$$\mu(t) = a_1 - \frac{a_2}{t^{a_3}}, \quad \sigma(t) = b_1 + \frac{b_2}{t^{b_3}} \quad (11)$$

The fitting of a model of the type of Eq. (11) requires the maximization of the Likelihood function

$$LF(a_1, a_2, a_3, b_1, b_2, b_3) = \prod_{i=0}^n \frac{1}{\sigma(t_i)\sqrt{2\pi}} e^{-[x_i - \mu(t_i)]^2 / 2\sigma(t_i)^2} \quad (12)$$

As in the case of nonlinear time series, the number of nonlinear terms grows with the sample size. We intend to study this Maximum Likelihood Estimation problem with the SFPM, comparing with other approaches for different sample sizes and varying the complexity of the probabilistic models.

## 4 Conclusions and Future Work

In our opinion the theory and results presented so far have the potential of considerably enrich the tools for global optimization. The characterization of optimization problems in terms of reliable probability densities may open the door to new insights into global optimization by the use of probabilistic and information – theoretic concepts. From a more practical standpoint, the proposed methodology may be implemented in a variety of ways in order to improve existing or construct new optimization algorithms.

We have so far studied unconstrained optimization problems, which are particularly relevant in parameter fitting of nonlinear models. The generalization to constrained problems, appears however to be straightforward. This is expected taking into account that the proposed method makes use of linear operations only. In this way, constraints may enter into Eq. (III) as additional nonlinear terms in the form of barriers, with no essential increment in computational cost.

These and related research lines are currently under study by the authors.

## Acknowledgements

D. P. and R. S. acknowledge financial support by CONACYT under scholarships 204271 and 204300. A. B. acknowledge partial financial support by CONACYT under grant J45702 – A, SEP under grant PROMEP/103.5/06/1584 and UANL.

## References

1. Berrones, A.: Generating Random Deviates Consistent with the Long Term Behavior of Stochastic Search Processes in Global Optimization. LNCS, vol. 4507, pp. 1–8. Springer, Heidelberg (2007)
2. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* 6, 721–741 (1984)
3. Grasman, J., van Herwaarden, O.A.: *Asymptotic Methods for the Fokker–Planck Equation and the Exit Problem in Applications*. Springer, Heidelberg (1999)
4. Haykin, S.: *Neural Networks: a Comprehensive Foundation*. Prentice Hall, New Jersey (1999)
5. Kantz, H., Schreiber, T.: *Nonlinear Time Series Analysis*. Cambridge University Press, Cambridge (2004)
6. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by Simulated Annealing. *Science* 220, 671–680 (1983)
7. Ott, E.: *Chaos in Dynamical Systems*. Cambridge University Press, Cambridge (1993)
8. Pardalos, P.M., Schoen, F.: Recent Advances and Trends in Global Optimization: Deterministic and Stochastic Methods. In: DSI 1–2004. Proceedings of the Sixth International Conference on Foundations of Computer–Aided Process Design, pp. 119–131 (2004)
9. Parsopoulos, K.E., Vrahatis, M.N.: Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural Computing* 1, 235–306 (2002)

10. Pavlidis, N.G., Plagianakos, V.P., Tasoulis, D.K., Vrahatis, D.K.: Human Designed Vs. Genetically Programmed Differential Evolution Operators IEEE Congress on Evolutionary Computation, pp. 1880–1886. IEEE Computer Society Press, Los Alamitos (2006)
11. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C++, the Art of Scientific Computing. Cambridge University Press, Cambridge (2005)
12. Risken, H.: The Fokker–Planck Equation. Springer, Heidelberg (1984)
13. Schweitzer, F.: Brownian Agents and Active Particles: Collective Dynamics in the Natural and Social Sciences. Springer, Berlin (2003)
14. Suykens, J.A.K., Vandewalle, J.: Artificial Neural Networks for Modelling and Control of Non-Linear Systems. Springer, Berlin (1996)
15. Suykens, J.A.K., Verrelst, H., Vandewalle, J.: On-Line Learning Fokker–Planck Machine. Neural Processing Letters 7(2), 81–89 (1998)
16. Van Kampen, N.G.: Stochastic Processes in Physics and Chemistry. North-Holland, Amsterdam (1992)

# From Horn Strong Backdoor Sets to Ordered Strong Backdoor Sets

Lionel Paris<sup>1</sup>, Richard Ostrowski<sup>1</sup>, Pierre Siegel<sup>1</sup>, and Lakhdar Saïs<sup>2</sup>

<sup>1</sup> LSIS CNRS UMR 6168

Université de Provence

13013 Marseille, France

<sup>2</sup> CRIL CNRS FRE 2499

Université d'Artois

62307 Lens, France

{lionel.paris, richard.ostrowski, pierre.siegel}@lsis.org,  
lakhdar.sais@cril.fr

**Abstract.** Identifying and exploiting hidden problem structures is recognized as a fundamental way to deal with the intractability of combinatorial problems. Recently, a particular structure called (strong) backdoor has been identified in the context of the satisfiability problem. Connections has been established between backdoors and problem hardness leading to a better approximation of the worst case time complexity. Strong backdoor sets can be computed for any tractable class. In [1], a method for the approximation of strong backdoor sets for the Horn-Sat fragment was proposed. This approximation is realized in two steps. First, the best Horn renaming of the original CNF formula, in term of number of clauses, is computed. Then a Horn strong backdoor set is extracted from the non Horn part of the renamed formula. in this article, we propose computing Horn strong backdoor sets using the same scheme but minimizing the number of positive literals in the non Horn part of the renamed formula instead of minimizing the number of non Horn clauses. Then we extend this method to the class of ordered formulas [2] which is an extension of the Horn class. This method insure to obtain ordered strong backdoor sets of size less or equal than the size of Horn strong backdoor sets (never greater). Experimental results show that these new methods allow to reduce the size of strong backdoor sets on several instances and that their exploitation also allow to enhance the efficiency of satisfiability solvers.

## 1 Introduction

Propositional satisfiability (SAT) is the problem of deciding whether a Boolean formula in conjunctive normal form (CNF) is satisfiable. SAT is one of the most studied NP-Complete problems because of its theoretical and practical importance. Encouraged by the impressive progress in practical solving of SAT, various applications (planning, formal verification, equivalency and logical circuits simplifications, etc.) are encoded and solved using SAT. Most of the best complete solvers are based on the backtrack search algorithm called Davis Logemann

Loveland (DPLL) procedure [3]. Such a basic algorithm is enhanced with many important pruning techniques such as learning, extended use of Boolean constraints propagation, preprocessing, symmetries detection, etc.

The other important class of satisfiability algorithms concerns local search based methods. These techniques can not prove unsatisfiability, since search space is explored in a non systematic way (*e.g.* [4,5]). However, impressive performance are obtained on hard and large satisfiable instances (including random instances).

Recently, several authors have focused on detecting possible hidden structures inside SAT instances (*e.g.* backbones [6], backdoors [7,8], equivalences [9] and functional dependencies [10]), allowing to explain and improve the efficiency of SAT solvers on large real-world instances. Other important theoretical results like heavy tailed phenomena [7] and backbones [11] were also obtained leading to a better understanding of problem hardness. The efficiency of satisfiability solvers can be in part explained using these results.

In this paper, we focus our attention on the strong backdoor sets computation problem. Let us recall that a set of variables forms a backdoor for a given formula if there exists an assignment to these variables such that the simplified formula can be solved in polynomial time. Such a set of variables is called a strong backdoor, if any assignment to these variables leads to a tractable sub-formula.

Computing the smallest strong backdoor is a NP-hard problem. Approximating (in polynomial time) the “smallest” strong backdoor set is an interesting and important research issue. Previous works have addressed this issue [10]. For example, the approach proposed in [10], try to recover a set of gates (Boolean functions) from a given CNF, then heuristics are used to determine a cycle cut-set from the graph representation of the new Boolean formula. A strong backdoor set made of both the independent variables and the variables of the cycle cut-set is then generated. However, on many problems where no gates are recovered, the strong backdoor cover the whole set of the original variables. Other approaches have been proposed using different techniques such as adapted systematic search algorithm [8,11]). Recently, in [12] an enhanced concept of sub-optimal reverse Horn fraction of CNF formula was introduced and an interesting correlation is observed with the satisfiability and the performances of SAT solvers on fixed density random 3-SAT instances. In [13], the relation between the search complexity of unsatisfiable random 3-SAT formulas and the sizes of unsatisfiable cores and strong backdoors were highlighted.

There are two major goals in this paper. The first one consists in designing a poly-time approach for computing strong backdoor sets of minimal size. This approach is based on the work of [1], for which we propose a new heuristic allowing to significantly reduce the size of the computed strong backdoor sets, considering the tractable class of Horn formulas. Then we extend this process to another tractable class for SAT, which is a natural extension of the Horn formulas: ordered formulas [2]. We show that thanks to the definition of ordered formula, we are sure to compute strong backdoor sets considering ordered formulas smaller or equal than those considering Horn formulas.

The paper is organized as follows. First we give preliminary definitions linked to the paper context and recall the method used in [1]. Then we present our new heuristic, that guides the local search, leading to Horn strong backdoor sets. In part 4, we present the extension of this approach to the tractable class of ordered formulas. Then we present some experimental results showing the interest of our method for computing strong backdoor sets. Finally, concerning the exploitation of the computed strong backdoor sets for solving instances, we present some experimental results showing that our method can significantly enhance one of the state of the art SAT solver. To conclude, the scope of these results is discussed and some paths for future works are given.

## 2 Preliminary Definitions and Notations

Let  $\mathcal{B}$  be a Boolean (*i.e.* propositional) language of formulas built in the standard way, using usual connectives ( $\vee, \wedge, \neg, \Rightarrow, \Leftrightarrow$ ) and a set of propositional variables. A *CNF formula*  $\Sigma$  is a set (interpreted as a conjunction) of *clauses*, where a clause is a set (interpreted as a disjunction) of *literals*. A literal is a positive or negated propositional variable. Let us recall that any Boolean formula can be translated to CNF using linear Tseitin encoding [14]. The size of CNF  $\Sigma$  is defined by  $\sum_{c \in \Sigma} |c|$  where  $|c|$  is the number of literals in  $c$ . We define  $c^+$  (resp.  $c^-$ ) as the set of positive (resp. negative) literals of a clause  $c$ . A *unit* (resp. *binary*) clause is a clause of size 1 (resp. 2). A *unit literal* is the unique literal of a unit clause. A unit clause  $\mathcal{C} = \{l\}$  is said to be positive if its unit literal  $l$  is positive ( $l \in \mathcal{V}$ ). We note  $nbVar(\Sigma)$  (resp.  $nbCla(\Sigma)$ ) the number of variables (resp. clauses) of  $\Sigma$ .  $\mathcal{V}(\Sigma)$  (resp.  $\mathcal{L}(\Sigma)$ ) is the set of variables (resp. literals) occurring in  $\Sigma$ . A set of literals  $S \subset \mathcal{L}(\Sigma)$  is consistent iff  $\forall l \in S, \neg l \notin S$ . Given a set of literals  $S$ , we define  $\sim S = \{\neg l | l \in S\}$ . A literal  $l \in \mathcal{L}(\Sigma)$  is called monotone if  $\neg l \notin \mathcal{L}(\Sigma)$ . We note  $\mathcal{V}^+(\Sigma)$  (resp.  $\mathcal{V}^-(\Sigma)$ ) as the set of variables occurring in  $\Sigma$  at least one time with positive (resp. negative) polarity. For each literal  $l$ ,  $Occ_{\Sigma}(l)$  represents the set  $\{C \in \Sigma | l \in C\}$  *i.e.* the set of clauses of  $\Sigma$  in witch literal  $l$  appears. We will simply write  $Occ(l)$  if no confusion can be made.

A *truth assignment* of a Boolean formula  $\Sigma$  is an assignment of truth values  $\{true, false\}$  to its variables. A variable  $x$  is satisfied (resp. falsified) under  $I$  if  $I[x] = true$  (resp.  $I[x] = false$ ). A truth assignment  $I$  can be represented as a set of literals. A literal  $l \in I$  (resp.  $\neg l \in I$ ) if  $I[l] = true$  (resp.  $I[l] = false$ ). When considering partial assignments, in witch some variables are not yet instantiated, those uninstantiated variables does not occur in the interpretation, neither in positive polarity, nor in negative one. We define  $I(\Sigma)$  as the formula simplified by  $I$ . Formally  $I(\Sigma) = \{C \setminus l | C \in \Sigma, I \cap C = \emptyset, \neg l \in I\}$ . A *model* of a formula  $\Sigma$  is a truth assignment  $I$  such that  $I(\Sigma) = \{\}$ . Accordingly, SAT consists in determining if the formula admits a model, or not.

We define a *renaming*  $R$  of  $\mathcal{V}(\Sigma)$  as the application of  $\mathcal{V}(\Sigma)$  in  $\{true, false\}$ .  $l_R$  is the literal corresponding to the renaming of literal by  $R$ . If  $R(\mathcal{V}(l)) = true$ ,

then  $l_R = \neg l$ , else  $l_R = l$ . The renamed variables are those which have the value *true* in  $R$ .

The idea proposed in [1], consists in computing a Horn strong backdoor set with a stochastic approach. In a first phase, a renaming is calculated in order to minimize the non Horn part of the formula. With this intention, the following definitions are given in [1] :

**Definition 1 (Renamed Formula).** *Let  $\Sigma$  be a CNF formula and  $R$  be a renaming of  $\mathcal{V}(\Sigma)$ . We define the renamed formula  $\Sigma_R$  as the formula obtained by substituting, for each variable  $x$  such that  $R(x) = \text{true}$ , every occurrences of  $x$  (resp.  $\neg x$ ) by  $\neg x$  (resp.  $x$ ).  $x$  is then renamed in  $\Sigma$ . When  $\Sigma_R$  is a Horn formula,  $R$  is said to be a Horn renaming of  $\Sigma$ .*

*We say that a literal  $l$  is positive in  $c$  for  $R$  if  $l \in c$  and  $\neg l \in R$ , or if  $\neg l \in c$  and  $l \in R$ . We denote it by  $isPos(l, c, R)$ . This literal is negative in  $c$  for  $R$  if  $l \in c$  and  $l \in R$ , or if  $\neg l \in c$  and  $\neg l \in I$ . This is denoted  $isNeg(l, c, R)$ .*

*The number of positive literals (resp. negative) in  $c$  for  $R$  is denoted by  $nbPos(c, R)$  (resp.  $nbNeg(c, R)$ ).*

*The total number of positive literals appearing in the non Horn part of  $\Sigma$  for the renaming  $R$  is denoted  $nbPosTot(\Sigma, R)$ .*

This allows to define the notion of *Horn renamability*.

**Definition 2 (Horn Renamability).** *Let  $\Sigma$  be a CNF formula and  $R$  be a renaming. A clause  $c \in \Sigma$  is said to be Horn renamed by  $R$  (denoted  $h\_ren(c, R)$ ) if  $nbPos(c, R) \leq 1$  i.e.  $c$  contains at most one positive renamed literal for  $R$ ; otherwise  $c$  is said to be Horn falsified by  $R$  (denoted  $h\_fal(c, R)$ ).*

*We denote  $nbHorn(\Sigma, R)$  as the number of clauses of  $\Sigma$  Horn renamed by  $R$ .*

The choice of the best renaming is made according to the following Objective Function :

**Definition 3 (Objective Function).** *Let  $\Sigma$  be a CNF formula,  $x \in \mathcal{V}(\Sigma)$ ,  $R$  be a renaming and  $R_x$  the renaming obtained from  $R$  by flipping the truth value of  $x$ . We define  $h\_breakCount(x, R) = |\{c | h\_ren(c, R), h\_fal(c, R_x)\}|$  and  $h\_makeCount(x, R) = |\{c | h\_fal(c, R), h\_ren(c, R_x)\}|$ . We define  $h\_score(x, R) = h\_makeCount(x, R) - h\_breakCount(x, R)$ .*

Once the renaming is computed, a greedy method is used in order to build a strong backdoor set. The principle is to choose the variable appearing the greatest number of times in the non Horn clauses of the formula, but only considering the positive literals of these clauses. The experimental results showed the usefulness of computing such sets. On the one hand, this allows to refine the worst case time complexity for the considered instances, and on the other hand, from a resolution point of view, many improvements in terms of time and number of nodes were noted.



### 3 A New Objective Function for Horn Formulas

In this part, we present a new objective function for the computation of the best Horn renaming. The approach used to compute the Horn strong backdoor, after the renamed formula is obtained remains the same.

As opposed to the one proposed in [11], this new objective function takes into account the size of the non Horn sub-formula. More precisely, it considers the number of positive literals occurring in the non Horn part of the formula. This objective function tries to find a renaming that minimize this size. It seems reasonable to think that if we succeed in reducing the total number of literals occurring in the non Horn part of the formula, even if the total number of non Horn clause is greater, the number of variables included in the strong backdoor set should decrease too, because it is built upon this reduced set of literals.

To implement this new objective function, we only have to redefine the *Horn Break count* and *Horn Make count* in order to allow the *h\_score* function of algorithm computing the best renaming to match the new objective function.

**Definition 4 (Min {Break/Make} Count).** *Let  $\Sigma$  be a CNF formula,  $x \in \mathcal{V}(\Sigma)$ ,  $R$  be renaming and  $R_x$  be the renaming obtained from  $R$  by flipping the truth value of  $x$ .*

*Let  $h\_Break_1 = \{c \in \Sigma \mid nbPos(c, R) = 1 \text{ and } isNeg(x, c, R)\}$ ,*

*$h\_Break_2 = \{c \in \Sigma \mid nbPos(c, R) > 1 \text{ and } isNeg(x, c, R)\}$ ,*

*$h\_Make_1 = \{c \in \Sigma \mid nbPos(c, R) = 2 \text{ and } isPos(x, c, R)\}$*

*and  $h\_Make_2 = \{c \in \Sigma \mid nbPos(c, R) > 2 \text{ and } isPos(x, c, R)\}$ .*

*The different counters are defined as follows:*

*$hM\_breakCount(x, R) = 2 \times |h\_Break_1| + |h\_Break_2|$  and*

*$hM\_makeCount(x, R) = 2 \times |h\_Make_1| + |h\_Make_2|$ .*

*Then we define  $hM\_score(x, R) = hM\_makeCount(x, R) - hM\_breakCount(x, R)$*

The set  $h\_Break_1$  corresponds to the set of clauses which are Horn renamed by  $R$  and which are not by  $R_x$ . This means that each of these clauses contains *two* positive literals for  $R_x$ , and that we have to add *two* literals for each clause of  $Break_1$  to  $nbPosTot(\Sigma, R)$  if we replace  $R$  by  $R_x$ . (i.e.  $nbPosTot(\Sigma, R_x) = 2 + nbPosTot(\Sigma, R)$  for each of these clauses). The set  $h\_Break_2$  corresponds to the set of clauses which are not Horn renamed by  $R$  and in which  $x$  is negative for  $R$ . These clauses are not Horn renamed by  $R_x$  either, and each of them contains *one* more positive literal for  $R_x$  than for  $R$ . We have to add *one* literal to  $nbPosTot(\Sigma, R)$  for each of these clauses if we replace  $R$  by  $R_x$ .

On the other hand, the set  $h\_Make_1$  corresponds to the set of clauses which are not Horn renamed by  $R$  and which are Horn renamed for  $R_x$ . This means that each of these clauses contains *two* positive literals for  $R$  that we have to withdraw to  $nbPosTot(\Sigma, R)$  when switching from  $R$  to  $R_x$ . And lastly, the set  $h\_Make_2$  corresponds to the set of clauses which are neither Horn renamed for  $R$ , nor for  $R_x$ , and in which  $x$  is positive for  $R$ . We have to remove *one* literal to  $nbPosTot(\Sigma, R)$  for each of these clauses when replacing  $R$  by  $R_x$ .

We just have to replace the function  $h\_score$  (line 17) by the function  $hM\_score$  to compute the renaming minimizing the number of positive literals in the clauses which are not Horn.

## 4 Ordered Formulas

In this section, we show how it is possible to compute strong backdoor sets for another tractable class of SAT: the class of ordered formulas, introduced by Benoist and Hébrard in 1999 [2].

### 4.1 Description

The class of ordered formulas can naturally be seen as an extension of the class of Horn formulas. It is based on the notion of free and linked literals [2].

**Definition 5 (Free/Linked Literals).** *Let  $C \in \Sigma$  be a clause and  $l \in C$  be a literal. We say that  $l$  is linked in  $C$  (with respect to  $\Sigma$ ) if  $Occ(-l) = \emptyset$  or there exists  $t \in C (t \neq l)$  such that  $Occ(-l) \subseteq Occ(-t)$ . If  $l$  is not linked in  $C$ , we say  $l$  is free in  $C$ .*

Ordered formulas are defined as follows [2]:

**Definition 6 (Ordered Formulas).** *A CNF formula  $\Sigma$  is ordered if each clause  $C \in \Sigma$  contains at most one free positive literal in  $C$ .*

By recalling that a CNF formula is a Horn formula if each of its clause contains at most one positive literal, we immediately see that any Horn formula is inevitably ordered. This supports the idea that the class of ordered formulas is an extension of the class of Horn formulas.

We know that if  $\Sigma$  is a Horn formula and does not contain any positive unit clause, then  $\Sigma$  is satisfiable. This property still holds when  $\Sigma$  is ordered. For ordered formula, we also have the following proposition (proven in [2]):

**Proposition 1.** *if  $\Sigma$  is ordered and does not contain any positive unit clause  $C = \{x\}$  such that  $x$  is free in  $C$ , then  $\Sigma$  is satisfiable.*

Besides, it is shown in [2] that an ordered formula is stable under unit propagation *i.e.* an ordered formula remains ordered whatever the realized unit propagations. Thus, added to proposition 1, we can see that the satisfiability problem for ordered formulas can be solved in linear time.

Lastly, like for Horn formulas, the class of ordered formulas can be extended by the class of ordered renameable formulas. There exists a polynomial algorithm for deciding whether a formula is ordered renameable or not, and another polynomial one for solving them [2].

Thus, we can try to calculate ordered strong backdoor sets by using the same scheme as for Horn strong backdoor sets *i.e.* starting from an original CNF formula, find a renaming minimizing the non ordered part and extract strong backdoor sets from the non ordered part of the renamed formula.

## 4.2 Approximating Maximal Ordered Renameable Sub-formula

As in the case of Horn formulas, there exists a polynomial algorithm for deciding of the ordered renameability of a CNF formula ([2]), and just like Horn case, if the formula is not ordered renameable, computing the maximal ordered renameable sub-formula is NP-hard. Indeed, in the worst case, if any literal is linked, the class of ordered formulas is equivalent to the class of Horn formulas.

The following proposition allows us to exploit the same algorithm to approximate the maximal ordered renameable sub-formula:

**Proposition 2.** *Let  $\Sigma$  be a CNF formula and  $c \in \Sigma$  be a clause of  $\Sigma$ . Let  $l$  and  $t$  be to literals of  $c$ . Let  $R$  be a renaming of the variables of  $\Sigma$  ( $\mathcal{V}(\Sigma)$ ). If  $l$  is linked to  $t$  in  $c$  with respect to  $\Sigma$ , then  $l_R$  is still linked to  $t_R$  in  $c_R$  with respect to  $\Sigma_R$ .*

*Proof.* It is sufficient to see that if a variable  $v \in \mathcal{V}(\Sigma)$  is renamed in  $R$ , then, for its associated literals  $v$  and  $\neg v$ , we have  $Occ_{\Sigma}(v) = Occ_{\Sigma_R}(v_R) = Occ_{\Sigma_R}(\neg v)$  and  $Occ_{\Sigma}(\neg v) = Occ_{\Sigma_R}(\neg v_R) = Occ_{\Sigma_R}(v)$ .

This proposition ensures us that the link between two literals is stable under renaming. Thus, for a given CNF formula, it is sufficient to compute only once a table containing the linked literals of the entire formula, and at any time, we just have to look at this table to know if a literal is linked or not, independently of the current renaming.

Here again, two criteria for computing the best renaming are possible: minimizing the number of non ordered clauses or minimizing the number of free positive literals in the non ordered clauses.

We need the following definitions before describing the two different objective functions that we propose.

**Definition 7 (Free Renamed Literal).** *Let  $\Sigma$  be a CNF formula and  $R$  be a renaming of  $\mathcal{V}(\Sigma)$ . We say that a literal  $l$  is free and positive in  $c$  for  $R$  if  $isPos(l, c, R)$  and  $l$  is not linked in  $c$  with respect to  $\Sigma$ . This is denoted  $isPos\&Free(l, c, R)$ . A literal  $l$  is free and negative in  $c$  for  $R$  is  $isNeg(l, c, R)$  and  $l$  is not linked in  $c$  with respect to  $\Sigma$ . This is denoted  $isNeg\&Free(l, c, R)$ .*

*We denote by  $nbPos\&Free(c, R)$  (resp.  $nbNeg\&Free(c, R)$ ) the number of positive (resp. negative) and free literal in  $c$  for  $R$ .*

*The total number of positive and free literals appearing in the non ordered part of  $\Sigma$  for  $R$  is denoted  $nbPos\&FreeTot(\Sigma, R)$ .*

This definition allows us to define the notion of *ordered renameability*:

**Definition 8 (Ordered Renameability).** *Let  $\Sigma$  be a CNF formula and  $R$  be a renaming. A clause  $c \in \Sigma$  is said to be ordered renamed by  $R$  (denoted  $o\_ren(c, R)$ ) if  $nbPos\&Free(c, R) \leq 1$  i.e.  $c$  contains at most one free positive literal for  $R$ ; else she is said to be ordered falsified by  $R$  (denoted  $o\_fal(c, R)$ ).*

*We denote  $nbOrd(\Sigma, R)$  as the number of clauses of  $\Sigma$  ordered renamed by  $R$ .*

Now we just have to define the different counters of gain (*Makecount*) and loss (*Breakcount*), and the associated *score* functions for both objective functions.

In order to minimize the number of non ordered clauses:

**Definition 9** (**{Break/Make}Count**). *Let  $\Sigma$  be a CNF formula,  $x \in \mathcal{V}(\Sigma)$ ,  $R$  be a renaming and  $R_x$  the renaming obtained from  $R$  by flipping the truth value of  $x$ . We define  $o\_breakCount(x, R) = |\{c | o\_ren(c, R), o\_fal(c, R_x)\}|$  and  $o\_makeCount(x, R) = |\{c | o\_fal(c, R), o\_ren(c, R_x)\}|$ . We define  $o\_score(x, R) = o\_makeCount(x, R) - o\_breakCount(x, R)$*

In order to minimize the number of free positive literals in the non ordered clauses:

**Definition 10.** *Let  $\Sigma$  be a CNF formula,  $x \in \mathcal{V}(\Sigma)$ ,  $R$  be a renaming and  $R_x$  the renaming obtained from  $R$  by flipping the truth value of  $x$ .*

*Let  $o\_Break_1 = \{c \in \Sigma | nbPos\&Free(c, R) = 1 \text{ and } isNeg\&Free(x, c, R)\}$ ,*

*$o\_Break_2 = \{c \in \Sigma | nbPos\&Free(c, R) > 1 \text{ and } isNeg\&Free(x, c, R)\}$ ,*

*$o\_Make_1 = \{c \in \Sigma | nbPos\&Free(c, R) = 2 \text{ and } isPos\&Free(x, c, R)\}$*

*and  $o\_Make_2 = \{c \in \Sigma | nbPos\&Free(c, R) > 2 \text{ and } isPos\&Free(x, c, R)\}$ . The counters are defined as follows:*

*$oM\_breakCount(x, R) = 2 \times |o\_Break_1| + |o\_Break_2|$  and*

*$oM\_makeCount(x, R) = 2 \times |o\_Make_1| + |o\_Make_2|$ .*

*Then we define  $oM\_score(x, R) = oM\_makeCount(x, R) - oM\_breakCount(x, R)$ .*

The sets *o\_Break* and *o\_make* are similar to the sets *h\_Break* and *h\_make* (definition 4) respectively, with the additional constraint that  $x$  must be free at each step, because all linked literals are ignored.

For a CNF formula, by replacing  $nbHorn(\Sigma)$  by  $nbOrd(\Sigma)$  and *h\_score* by *o\_score*, we obtain an algorithm to calculate a renaming of  $\Sigma$  minimizing the number of non ordered renamed clauses. This renaming will be called *Ordered\_Min-Clause (OMC)*.

By replacing *o\_score* by *oM\_score* in it, we obtain an algorithm to calculate a renaming minimizing the number of free positive literals in the non ordered renamed clauses. This renaming will be called *Ordered\_Min-Literals (OML)*.

### 4.3 Computing Ordered Strong Backdoor Sets

One more time, the problem of computing ordered strong backdoor sets of minimal size is NP-hard. Then we will use the same process as for Horn strong backdoor sets *i.e.* for a CNF formula, compute a sub-set of minimal size of the free positive literals occurring in the non ordered sub-formula, such that when removed the literal of this set, the remaining of the formula is ordered. However, this is valid only if the withdrawal of these literals, by assigning them to true or false, does not render free other literals of the formula which were linked. The following proposition ensures us that this is the case:

**Proposition 3.** *Let  $\Sigma$  be a CNF formula. Let  $\mathcal{U}$  be a set of unit clauses over  $\mathcal{V}$ . Let  $I$  be a truth assignment such that  $I = \mathcal{V}(\mathcal{U})$ . Let  $C \in \Sigma$ ,  $l \in C$  such*

that  $l$  is linked in  $C$  with respect to  $\Sigma$ , then  $l$  is also linked in  $C'$  with respect to  $\Sigma' = I(\Sigma)$  ( $C' = I(C)$ ); or  $C$  is removed from  $\Sigma'$  (because she is satisfied).

*Proof.* Because of the fact that  $l$  is linked, two cases are possible:

- $Occ_{\Sigma}(\neg l) = \emptyset$ : in this case, it is obvious that  $Occ_{\Sigma'}(\neg l) = \emptyset$ , and  $l$  is still linked in  $\Sigma'$  in each clause where it appears.
- $Occ_{\Sigma}(\neg l) \neq \emptyset$ :  $\exists t \in C$  such that  $Occ_{\Sigma}(\neg l) \subseteq Occ_{\Sigma}(\neg t)$ . Here, three cases are possible:
  - $t \in \mathcal{U}$ : in this case, the clause  $C$  is satisfied in  $\Sigma'$ .
  - $\neg t \in \mathcal{U}$ : in this case, each clause containing  $\neg t$  is satisfied (removed) in  $\Sigma'$ . Because  $Occ_{\Sigma}(\neg l) \subseteq Occ_{\Sigma}(\neg t)$ , every clauses containing  $\neg l$  are also satisfied in  $\Sigma'$ . Thus  $Occ_{\Sigma'}(\neg l) = \emptyset$  and  $l$  is still linked in  $C'$  with respect to  $\Sigma'$ .
  - $t \notin \mathcal{U}$  and  $\neg t \notin \mathcal{U}$ : let  $S$  be the set of clauses of  $\Sigma$  satisfied (removed) in  $\Sigma'$ . We have  $Occ_{\Sigma'}(\neg t) = Occ_{\Sigma \setminus S}(\neg t)$  and  $Occ_{\Sigma'}(\neg l) = Occ_{\Sigma \setminus S}(\neg l)$ . Because  $Occ_{\Sigma}(\neg l) \subseteq Occ_{\Sigma}(\neg t)$ , we have  $Occ_{\Sigma \setminus S}(\neg l) \subseteq Occ_{\Sigma \setminus S}(\neg t)$ . Then  $Occ_{\Sigma'}(\neg l) \subseteq Occ_{\Sigma'}(\neg t)$ , and  $l$  is still linked in  $C'$  with respect to  $\Sigma'$ .

With this proposition, we are ensured of the stability of linking under assignment *i.e.* in a CNF formula, each linked literal in a clause remains linked in the corresponding clause of the simplified formula with any assignment. Thus, we can compute an ordered strong backdoor set  $S$  of minimal size as follows: for any CNF formula  $\Sigma$ , we consider only the non ordered part  $\psi$  of  $\Sigma$ . We build  $S$  by adding in it variables for which the positive occurrences occur in  $\mathcal{V}^+(\psi)$ . These variables are then removed from  $\psi$ , and we go on iteratively until  $\psi$  become ordered. At each step, the variable corresponding to the positive and free literal with the greatest number of occurrences in  $\psi$  is chosen. We use the same algorithm in which we add the constraint that the chosen variables must be free.

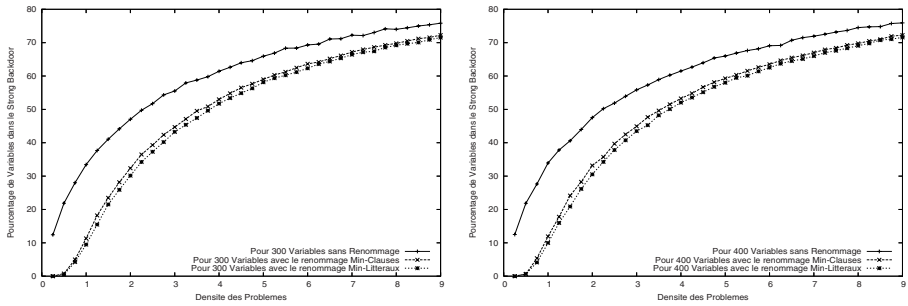
The computed ordered strong backdoor sets have a size in the worst case equal to the those computed for Horn formulas. Indeed, in the case where no literal is linked, the Horn strong backdoor sets and ordered strong backdoor sets are identical, but as soon as any literal is linked, as we can exclude all linked literals, the final size of the ordered strong backdoor sets can only be smaller than for the one of Horn strong backdoor.

## 5 Experimentations

### 5.1 Strong Backdoor Computation

First, we tested this method for computing strong backdoor sets on randomly generated 3-SAT instances. We experimented instances with 300 and 400 variables. For each of them, we applied a variation of the ratio  $\frac{\#C}{\#V}$  from 0.25 to 9, and for each ratio, we generated 50 instances and computed the average number of variables occurring in the Horn strong backdoor set according to the three following schemes:

1. we considered the original formula (without any renaming) and computed the Horn strong backdoor using the greedy algorithm.



**Fig. 1.** Size of the Strong Backdoor Sets for Random 3-SAT Instances

2. we considered the formula renamed by the best renaming obtained according to the criterion of the number Horn renamed clauses, and computed the Horn strong backdoor using the greedy algorithm. This renaming will be called *Horn\_Min-Clause* renaming (*HMC*).
3. we considered the formula renamed by the best renaming obtained according to the criterion of the number of positive literals appearing in the non Horn part of the renamed formula, and computed the Horn strong backdoor using the greedy algorithm. This renaming will be called *Horn\_Min-Literal* renaming (*HML*).

We only report the results obtained with the Horn class, because they were identical to those obtained with the ordered class as soon as the ratio  $\frac{\#C}{\#V}$  exceeded 2.

Figure 1 recaps the results obtained for each class of instances. We notice that compared to the original formula, we obtain fare better results in term of size of the Horn strong backdoor set, independently of the selected renaming. Moreover, our new heuristic appends to be better than the one proposed in [1] in terms of size of the strong backdoor set.

Table 1 shows some results of our approach on instances issued from previous SAT competitions. For each instance, we give the size of the instance, ( $\#V$  and  $\#C$ ), the size of the Horn strong backdoor set obtained with the heuristic of [1] (*i.e.* using *HMC*), the size of the Horn strong backdoor set computed using the new heuristic (*i.e.* using *HML*), the size the ordered strong backdoor set com-

**Table 1.** Size of the strong backdoor sets on some real-world instances

<i>Instance</i>	$\#V$	$\#C$	$SB_{HMC}$	$SB_{HML}$	$SB_{OMC}$	$SB_{OML}$
fifo8_100	64762	176313	22933	21065	21203	19551
f2clk_50	34678	101319	13923	13028	13453	12582
ip50	66131	214786	30280	29301	29720	28886
ip38	49967	162142	22911	21894	22429	21496
ca256	4584	13236	2127	1949	1984	1834
ca128	2282	6586	1067	977	980	897
example2_gr_2pin_w6	3603	41023	2597	2702	2579	2633
too_large_gr_rcs_w9	4671	64617	3397	3024	3385	3000

puted using the former heuristic (*i.e.* using *OMC*) and the the size the ordered strong backdoor set computed using the knew heuristic (*i.e.* using *OML*). This results show clearly the interest of minimizing the number of positive literals appearing in the non Horn/ordered part before computing the strong backdoor set, rather than minimizing the number of non Horn/ordered clauses. Generally, the formula renamed by the renaming obtained with the new heuristic contains more non Horn/ordered clauses than with the previous one, but the size of the strong backdoor sets obtained with *HML* (resp. *OML*) are in the worst case equal to those obtained with *HMC* (resp. *OMC*). Compared to *HML*, we see that using ordered formulas (*i.e.* *OMC* and *OML*) instead of Horn formulas allows to obtain smaller strong backdoor sets for many instances. This highlights the following phenomenon: although we never find ordered (renameable) instances in practice, many instances encoding real applications contain a rather significant number of linked literals, as shown by the size of the computed ordered strong backdoor sets compared to that of Horn strong backdoor sets.

## 5.2 Practical Resolution

In this section, we present experimental results that we obtained by exploiting strong backdoor sets while solving SAT instances. In these experimentations, the SAT solver *Zchaff* [1] was forced to branch on the variables contained by the strong backdoor sets.

The first experimentations were realized on random instances all unsatisfiable, with a ratio  $\frac{\#C}{\#V} = 4.25$ . As explained in section 5.1, because the ratio is bigger than 2, the computed strong backdoor sets for both Horn and ordered formulas are identical. That is why in table 2 we do not precise for which class the strong backdoor sets have been calculated, but only the renaming used: *MC* when minimizing the number of clauses and *ML* when minimizing the number of literals. In this table, line gives the average data for 50 instances: the time needed (in sec) to solve the instance (Time), the number of nodes explored during search (Nodes) and the maximum depth of the search tree reached during the search (MxD). All these data are given for the three experimented methods. We can notice that from each point of view exploiting strong backdoor sets significantly improve the performances of *Zchaff*, and it seems that the bigger the size of the instance is, the better *ML* approach behaves compared to *MC* approach.

Then we experimented the resolution of instances issued from the previous SAT competitions using strong backdoor sets. Table 3 gives the results obtained

**Table 2.** ZChaff On Random Instances

#V	#C	Zchaff			Zchaff+SB <sub>MC</sub>			Zchaff+SB <sub>ML</sub>		
		Time(s)	Nodes	MxD	Time(s)	Nodes	MxD	Time(s)	Nodes	MxD
200	850	5,15	55610,76	28,26	4,18	47071,1	25,52	4,07	46215,26	25,88
250	1062	192,18	388970,5	34,18	133,21	328879,58	30,46	138,4	332056,52	30,62
300	1275	4499,91	2287975,5	39	3601,22	2016263,68	34,9	3302,74	1978685,62	34,7

<sup>1</sup> 2003 Version.

**Table 3.** ZChaff On Industrial Instances

Instance	#V	#C	S/U	Zchaff		Zchaff+SB <sub>HMC</sub>		Zchaff+SB <sub>HML</sub>		Zchaff+SB <sub>OMC</sub>		Zchaff+SB <sub>OML</sub>	
				Time (s)	Nodes	Time (s)	Nodes	Time (s)	Nodes	Time (s)	Nodes	Time (s)	Nodes
okgen-c1300-v650	650	1300	S	0	431	0	194	0	166	0	178	0	165
dp07u06	3193	8308	U	1,11	6990	1,6	6636	1,38	5532	1,29	5481	1,44	6113
dp03s03	637	1468	S	0	59	0	87	0	44	0	41	0	35
dp12s12	12065	33520	S	<i>time</i>	<i>out</i>	<i>time</i>	<i>out</i>	<i>time</i>	<i>out</i>	333,46	274302	2416,93	1291718
rand_net40-30-1	2400	7121	U	1,06	8973	1,7	8716	1,95	10464	1,68	8281	1,85	9821
rand_net40-40-10	3200	9521	U	137,57	311358	152,01	314143	151,21	297820	167,47	320576	137,8	285891
rand_net40-40-5	3200	9521	U	53,46	133375	93,78	175079	56,07	127578	78,85	158466	82,52	156776
rand_net70-25-1	3500	10361	U	113,36	228410	77,98	163133	49,6	126285	72,73	151790	164,66	267168
rand_net40-25-5	2000	5921	U	198,08	344425	200,5	329849	40,99	132043	164,02	296932	62	169776
rand_net60-30-5	3600	10681	U	1295,94	1249286	727,51	833383	1034,97	1085851	633,2	749447	1173,38	1124490
w10_70	32745	103556	U	178,49	137310	282,19	145002	427,46	167610	277	126026	511,79	201524
fifo8_100	64762	176313	U	74,12	151084	264,06	105492	258,47	108329	355,34	141962	344,99	136466
hanoi6	7086	78492	S	181,4	248892	1020,28	766762	110,5	182502	1462,21	825750	184,21	248646
hanoi5	1931	14468	S	<i>time</i>	<i>out</i>	913,19	859670	784,78	769299	626,06	757422	2915,62	1755395
jp38	49967	162142	U	3010,57	1082808	2233,59	650405	3077,5	808930	<i>time</i>	<i>out</i>	<i>time</i>	<i>out</i>
unif-c2600-v650	650	2600	S	10,76	83141	274,48	468181	0,43	7053	250,03	450142	11,48	76866
unif-c2450-v700	700	2450	S	0,02	755	0,01	302	0,03	649	0,05	895	0,05	995

on some instances, using strong backdoor sets computed for both class of Horn formulas and ordered formulas, and using both the renaming minimizing the number of clauses (*\*MC*) and the number of literals (*\*ML*). Columns Time, Nodes and MxD are identical to previous ones, and column *S/U* informs us about the satisfiability of the instance (*S* for satisfiable and *U* for unsatisfiable).

We can conclude that in some cases, the exploitation of strong backdoor sets improve either the resolution time, or the the size of the search space (number of nodes). Unfortunately, it seems hard to conclude that strong backdoor sets computed for a class are better than those computed for the other one. We see that it heavily depends on the considered instance.

## 6 Conclusions and Perspectives

We present in this article a new method for computing Horn strong backdoor. This method exploit a renaming of the original formula that minimize the number of positive literals contained in the non Horn clauses. This method allowed us to compute smaller Horn strong backdoor sets, although the number of non Horn clauses is slightly higher.

Then we propose an extension of this method to compute ordered strong backdoor sets. This method ensures us that the number of variables contained in the ordered strong backdoor sets is always smaller or equal than the one contained in Horn strong backdoor sets. It allows us to reduce a little bit more the size of the computed strong backdoor sets.

We experimented the exploitations of these strong backdoor sets for the resolution of random instances and instances issued from previous SAT competitions using Zchaff solver. We noticed that concerning random instances, the performances of Zchaff were significantly improve by the use of strong backdoor sets (more than 20% of gain in time for instances containing 300 variables). On the other hand, on instances issued from previous SAT competitions, great disparities were noted. But, in some cases, the exploitation of strong backdoor sets brings substantial improvements at least concerning the space of the search tree,



or concerning the time needed for the resolution. However, it is hard to claim that a tractable class produce more relevant strong backdoor sets than another with this approach between Horn and ordered formulas.

The main prospect for future work which we consider is the integration of strong backdoor sets for solving MAX-SAT problem. Despite no tractable class for the MAX-SAT problem has been identified, we feel that it is possible to use Horn strong backdoor sets to solve this problem.

## References

1. Paris, L., Ostrowski, R., Siegel, P., Saïs, L.: Computing and exploiting horn strong backdoor sets thanks to local search. In: ICTAI 2006. Proceedings of the 18th International Conference on Tools with Artificial Intelligence, Washington DC, United States, pp. 139–143 (2006)
2. Benoist, E., Hébrard, J.J.: Ordered formulas. In: Les cahiers du GREYC, CNRS - UPRES-A 6072 (search report). Number 14, Université de Caen - Basse-Normandie (1999)
3. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. *Communications of the ACM* 5, 394–397 (1962)
4. Selman, B., Kautz, H.A.: An empirical study of greedy local search for satisfiability testing (1993)
5. Mazure, B., Saïs, L., Grégoire, É.: Tabu search for SAT. In: AAI 1997/ IAAI 1997. Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference, pp. 281–285. AAAI Press, Stanford (1997)
6. Dubois, O., Dequen, G.: A backbone-search heuristic for efficient solving of hard 3-sat formulae. In: IJCAI 2001. Proceedings of the 17th International Joint Conference on Artificial Intelligence (2001)
7. Williams, R., Gomes, C., Selman, B.: On the connections between heavy-tails, backdoors, and restarts in combinatorial search. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, Springer, Heidelberg (2004)
8. Williams, R., Gomes, C., Selman, B.: Backdoors to typical case complexity. In: IJCAI 2003. Proceeding of Interational Joint Conference on Artificial Intelligence (2003)
9. Li, C.M.: Integrating equivalency reasoning into davis-putnam procedure. In: proceedings of Conference on Artificial Intelligence AAAI, pp. 291–296. AAAI Press, USA (2000)
10. Grégoire, E., Mazure, B., Ostrowski, R., Saïs, L.: Automatic extraction of functional dependencies. In: Hoos, H.H., Mitchell, D.G. (eds.) SAT 2004. LNCS, vol. 3542, pp. 122–132. Springer, Heidelberg (2005)
11. Kilby, P., Slaney, J., Thiebaut, S., Walsh, T.: Backbones and backdoors in satisfiability. In: AAAI 2005 (2005)
12. Maaren, H.V., Norden, L.V.: Correlations between horn fractions, satisfiability and solver performance for fixed density random 3-cnf instances. *Annals of Mathematics and Artificial Intelligence* 44, 157–177 (2005)
13. Lynce, I., Marques-Silva, J.P.: Hidden structure in unsatisfiable random 3-sat: An empirical study. In: Proc. of ICTAI 2004 (2004)
14. Tseitin, G.: On the complexity of derivations in the propositional calculus. In: Slesenko, H. (ed.) *Structures in Constructives Mathematics and Mathematical Logic, Part II*, pp. 115–125 (1968)

# G-Indicator: An M-Ary Quality Indicator for the Evaluation of Non-dominated Sets

Giovanni Lizárraga, Arturo Hernández, and Salvador Botello

Center for Research in Mathematics (CIMAT)

Department of Computer Science

giovanni@cimat.mx, artha@cimat.mx, botello@cimat.mx

**Abstract.** Due to the big success of the Pareto's Optimality Criteria for multi-objective problems, an increasing number of algorithms that use it have been proposed. The goal of these algorithms is to find a set of non-dominated solutions that are close to the True Pareto front. As a consequence, a new problem has arisen, how can the performance of different algorithms be evaluated? In this paper, we present a novel system to evaluate  $m$  non-dominated sets, based on a few assumptions about the preferences of the decision maker. In order to evaluate the performance of our approach, we build several test cases considering different topologies of the Pareto front. The results are compared with those of another popular metric, the S-metric, showing equal or better performance.

## 1 Introduction

In order to solve multi-objective optimization problems, many algorithms based on the Pareto's Optimality Criteria (*POC*) have been developed [5] [6] [7]. Instead of generating a single solution, these algorithms generate a set  $X$  of vector solutions  $x$  that approximate the Pareto set (*PS*). A PS is a non-dominated set (defined next), while the True Pareto Set is a PS that can not be improved by any means. A Pareto front ( $P^*$ ) is the projection of the PS over the space of decision functions  $F(X) = \{f_1(x), f_2(x), \dots, f_M(x)\}$ . The Pareto front, another set itself, is usually described as a surface in the objective space that represents the best trade-off possible between the objective functions. Hereafter in the article we must locate points, sets, vectors, and solutions in the space of the objective functions.

The main property of these sets is that none of its elements is dominated by another one, therefore, they are usually called *non-dominated sets* (*NS*) (a vector  $A$  dominates a vector  $B$  if  $A$  is better or equal to  $B$  in all their components and there is at least one component for which  $A$  is better than  $B$ ).

One of the most important difficulties about using the POC is how to compare the performance of different algorithms. Ignoring other factors (like the computational complexity), the performance is based on the result, in other words, an algorithm is as good as the NS it generates. So, it is necessary to have a criterium

to evaluate a NS. In order to create such a criteria we need to decide what we want from a NS. The most important characteristics of a NS are the following [4]:

*Convergence:* it refers to how near a NS is to  $P^*$ .

*Distribution:* we want a NS with a uniform spatial distribution, because it gives better information of the topology of  $P^*$ .

*Extension:* in order to approximate  $P^*$  as much as possible, it is important to know the range of its values.

Out of the three, convergence is considered to be the most important, because it is related with how optimal are the points in the NS. It is usual to consider distribution and extension as a single characteristic, and for the rest of the document we refer to them as distribution–extension. Considering convergence and dominance, Hansen and Jaszkiwicz [8] define the three following relationships between two NSs.

*Weak outperformance:* A weakly outperforms  $B$  ( $A O_W B$ ) if for every point  $b \in B$  there exists a point  $a \in A$  so that  $a$  dominates or is equal to  $b$  and there exists at least a point  $c \in A$  so that  $c \notin B$ .

*Strong outperformance:* A strongly outperforms  $B$  ( $A O_S B$ ) if for every point  $b \in B$  there exists a point  $a \in A$  so that  $a$  dominates or is equal to  $b$  and there exists at least a pair of points  $r \in A$  and  $s \in B$  such that  $r$  dominates  $s$ .

*Complete outperformance:* A completely outperforms  $B$  ( $A O_C B$ ) if for every point  $b \in B$  there exists a point  $a \in A$  so that  $a$  dominates  $b$ .

It is clear that  $O_C \subset O_S \subset O_W$ . These outperformance relations create a partial order between NSs, but they are not performance metrics themselves. However, they are useful to evaluate the robustness of comparison methods. Hansen and Jaszkiwicz [8] define the compatibility with an outperformance relation  $O$ , where  $O$  can be  $O_W$ ,  $O_S$  or  $O_C$ , as follows.

*Weak compatibility.* A comparison method  $R$  is weakly compatible with  $O$  if for two NSs  $A$  and  $B$ ,  $A O B$  implies that  $R$  will evaluate  $A$  as not worse than  $B$ .

*Compatibility.* A comparison method  $R$  is compatible with  $O$  if for two NSs  $A$  and  $B$ ,  $A O B$  implies that  $R$  will evaluate  $A$  as better than  $B$ .

The compatibility with outperformance relations is desirable because it makes the comparison methods more robust to misleading cases. Many performance measures have been proposed in the past [1] [2] [3] [8]. At the beginning, the approaches were focused on the three main characteristics of NSs (convergence, distribution, extension) and more recently, the focus is on the compatibility with outperformance relations.

There is a tendency to see the convergence and the distribution–extension as conflicting characteristics that are difficult to combine in a single metric. Some authors [10] have proposed to measure these characteristics individually and combine them in some way, in order to evaluate a NS. In this work we propose a performance measure that is  $O_C$  compatible, is very effective comparing the

distribution–extension, its computational complexity is moderate and makes intuitive decisions with respect to the quality of the NSs. We call it the G–Indicator and it is described in the next section.

## 2 G–Indicator

In this work we propose an  $m$ –ary performance measure for NSs, we call it the G–Indicator  $G$ . This measure assigns a real, positive number to each NS, the bigger the number, the better the NS. The value of the G is relative to the  $m$  sets we are comparing. If a new set is added or deleted, the value of G should be recalculated for the  $m+1$  or  $m-1$  sets. G has two main components, one for distribution–extension and another for convergence, the most important properties for a NS. First, we explain the distribution–extension component, then the convergence component, and finally, how to combine both to create the G–Indicator.

### 2.1 Distribution–Extension Component

The distribution–extension of a NS is an important characteristic, it is related with how well distributed are the vectors in the objective space. A good distribution–extension increases the chances that for any particular trade–off between objective functions  $TA$ , the user will find a solution  $TB \in \text{NS}$  that is similar to  $TA$ .

In our approach we consider the region around every element  $p$  of a NS that consists of all the points whose distance is inferior to a limit  $U$ , for example, the circle around  $p$  in Figure 1. Based on these ideas, we give the following definitions:

*Zone of influence of point  $p_i$  ( $I_{p_i}$ ).* It is the set of points whose distance from  $p_i$  is equal or less than a real, positive number  $U$ .

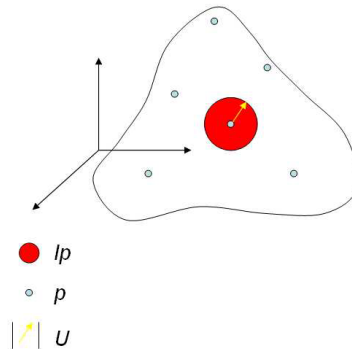
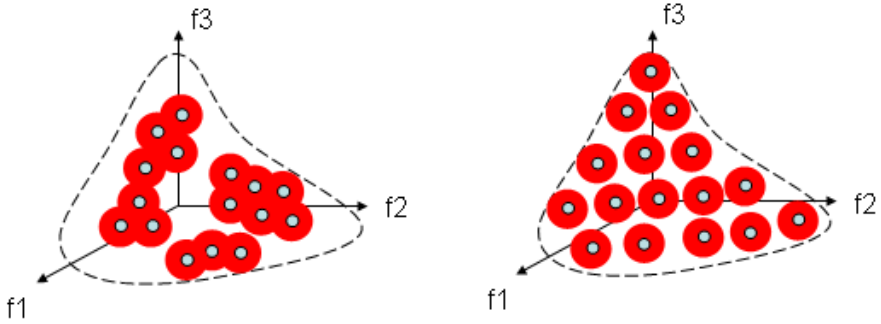


Fig. 1. An example of  $I_p$

*Zone of influence of the set  $S$  ( $IS$ ).* It is the union of the  $Ip_i$  for all  $p_i \in S$ .

In general, we refer to a zone of influence as  $I$ . Now, we review the behavior of  $I$  for a NS according to the configuration of its elements. In Figure 2, the dashed line represents the contour of the True Pareto front of a problem with three objective functions, the small circles are the elements  $p$  of the NS, and the big circles represent the  $Ip$ . If  $U$  is chosen wisely (we explain later how to choose  $U$ ), the zone of influence for the set on the left is small due to the bad distribution of its points. In contrast, the  $I$  for the set on the right is bigger because of the good distribution of its elements.



**Fig. 2.** The set on the left has poor dispersion. The set on the right has good dispersion.

The behavior of the  $I$  is exactly what we expect from a good distribution-extension measure. The value of  $IS$  is proportional to the distribution-extension of the  $S$ , so it is a good indicator. It is important to say that the  $IS$  is inversely proportional to the overlap between the  $Ip_i$  for  $p_i \in S$ . Sets of points with good distribution-extension have less overlap than those with bad distribution-extension. In conclusion, we use  $I$  as the distribution-extension component for this approach.

Calculating  $IS$  for a set  $S$  could be complex, especially for big sets in high dimensions. In order to reduce the complexity we propose an approximation of  $IS$  instead of its exact value.

**Computing  $IS$ .** Given a set of points  $S$  and a limit  $U$ .

1. For each  $p_i \in S$  find the nearest neighbors in all directions, to the right and to the left of  $p_i$ . For example, for a set  $S$  with 3 objective functions, for every  $p_i \in S$  choose the point that is nearest to  $p_i$  on the right and the nearest point on the left according to function 1, then the nearest point on the right and the nearest point on the left for function 2 and the same for function 3. A nearest neighbor is not allowed to appear more than once for the same  $p_i$ , so there will be at most  $2d$  nearest neighbors, where  $d$  is the number of objective functions. This definition of nearest neighbors and the procedure to find them is proposed by [1].

2. Next, calculate the distances  $D_{ij}$  for  $p_i$  and its  $j$ th nearest neighbor.
3. Then, calculate the mean  $D_i$  of the distances  $D_{ij}$ .
4. Next, calculate a radius  $r_i = D_i/2$ .
5. In order to avoid that  $r_i$  is bigger than  $U$ , make the following verification: if ( $r_i > U$ ) then make  $r_i = U$ .
6. Next, calculate:

$$Ip_i = r_i^{d-1} . \quad (1)$$

This value for  $r_i$  produces a small  $Ip$  for points with a lot of neighbors close to them, and a big  $Ip$  for isolated points. This way, the overlap is reduced and the IS is roughly calculated with the following formula:

$$IS = \frac{\sum_{i=1}^{|S|} Ip_i}{|S|} . \quad (2)$$

Equation (II) is a variation of the (hyper) area of a circle, we simply take out the constant factor. The Pareto front is usually described as an  $(n-1)$ -dimensional surface in a  $n$ -dimensional space, for example, it is a line in  $2d$  space. That is why we use the  $n-1$  power instead of the  $n$  power.

**The parameter  $U$ .** As we mentioned before, the overlap between  $Ip$  is very important to the good behavior of  $I$  as a dispersion-extension measure. At the same time, the overlap depends on the parameter  $U$ , so the numeric value of  $U$  must be chosen carefully. It is important to allow some level of overlap, so we can discriminate between sets with good and bad distribution, if  $U$  is too small the overlap will be zero and the value of  $I$  will not be related to the distribution of the points. We propose the following value of  $U$  for  $m$  non-dominated sets  $S_j$ :

$$U = 0.5 \frac{\sum_{j=1}^m \sum_{i=1}^{|S_j|} r_{ij}}{\sum_{j=1}^m |S_j|} . \quad (3)$$

where  $r_{ij}$  is the mean of the distances between  $p_i \in S_j$  and its nearest neighbors. This value of  $U$  produces, at least, a small amount of overlap in the  $Ip$  of the NS with the worst distribution. It is important to note that  $U$  depends on all the  $m$  sets, if a NS is added or eliminated, it must be recalculated.

**Scale and normalization.** A very important detail to consider is the scale of the objective functions. If an objective function has a bigger scale than the others, its influence will be more important. In order to avoid this we use the following normalization before the computation of  $I$ :

1. Take the union of the  $m$  sets,  $A = \bigcup_{i=1}^m S_i$ .
2. Take from  $A$  its non-dominated elements.  $A^* = ND(A)$ .
3. Find  $max_i$  and  $min_i$  as the maxima and minima value respectively, for the component  $i$  for all points  $p \in A^*$ .
4. Using  $max_i$  and  $min_i$  make a linear normalization of all points in all sets  $S$ .

We use this normalization because we consider it better to normalize with respect to the known Pareto front, because dominated points can have high values in their components introducing noise in the scale if they are used for the normalization.

### 2.2 Convergence Component

Convergence is the most important characteristic of a NS, so it is necessary for a performance measure to consider convergence in its evaluation mechanism.

Now, in order to simplify the explanation of the convergence component, we define a relationship between a group of NSs with another NS. It is an extension of the complete outperformance  $O_c$ .

*Complete Outperformance of a Set of NSs.* A set  $C = \{A_1, A_2, \dots, A_m\}$ , whose elements  $A_i$  are NS, completely outperforms a NS  $B$  ( $C O_c B$ ) if for every point  $b \in B$  exists a point  $a \in A_i$  for  $i \in \{1, 2, \dots, m\}$  so that  $a$  dominates  $b$ . If  $C$  does not completely outperforms  $B$ , we write it as  $C O'_c B$ .

Our convergence operator classifies the NSs by levels, based on the outperformance between every NS with the rest of the NSs. If we have  $m$  sets, the first level  $L_1$  includes those NSs  $A_k$  so that  $\{A_1, A_2, \dots, A_m\} O'_c A_k$ . The following levels include those NSs that are completely outperformed only by the previous levels. The pseudocode is as follows:

Given a set  $C = \{A_1, A_2, \dots, A_m\}$  where  $A_i$  is a NS and  $i \in \{1, 2, \dots, m\}$ .

1. Make  $j = 1$ .
2. Make  $L_j = \{\}$
3. Take from  $C$  those  $A_i$  such that  $C O'_c A_i$  and put them in  $L_j$ .
4. If  $C$  is not empty, make  $j = j + 1$  and return to step 2. Otherwise, end.

The  $A_i \in L_1$  are in the first level, the  $A_i \in L_2$  are in the second level and so on. If  $A \in L_j, B \in L_k$ , and  $j < k$  we consider  $A$  better than  $B$ . As an example, in Figure 3, there are five NSs,  $A, B, C, D$  and  $E$ . For this case, we have three levels of dominance, where  $L_1 = \{A\}$ ,  $L_2 = \{B, C\}$  and  $L_3 = \{E, F\}$ .

This convergence operator creates a partial order in the NSs and it is compatible with  $O_c$ , because if  $A O_c B$ , then  $A$  will be in a better level than  $B$ .

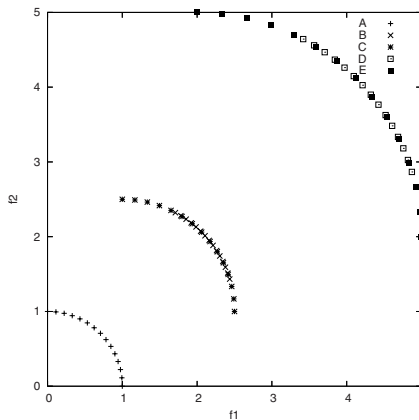


Fig. 3. Five NSs, three levels of dominance

### 2.3 Computing G-Indicator

Once we have defined the dispersion-extension and convergence operators, the procedure to calculate the G-indicator is as follows:

Given  $m$  non-dominated sets,  $A_1, A_2, \dots, A_m$ .

1. Normalize all sets as described before.
2. Classify all sets by levels of dominance (convergence operator).
3. For  $k = 1 : R$ , where  $R$  is the number of levels.
  - (a) For every  $A \in L_k$  eliminate all points  $p \in A$  dominated by (an)other point(s)  $q \in B$  for any  $B \in L_k$ .
  - (b) Calculate  $U$  based on all  $A \in L_k$ .
  - (c) Calculate the  $IA$  for all  $A \in L_k$ .
  - (d) Calculate a compensation  $D_k = \max(IA)$  for  $A \in L_k$ .
4. For  $k = 1 : R - 1$  where  $R$  is the number of levels.
  - (a) For all  $A \in L_k$  the value of the G-Indicator is:

$$G(A) = IA + \sum_{i=k+1}^R (D_i + \epsilon) . \quad (4)$$

where  $\epsilon$  is a real constant greater than zero.

5. For all  $A \in L_R$ ,  $G(A) = IA$ .

The compensation  $D$  and the constant  $\epsilon$  in formula (4) are necessary to keep the order between the dominance levels and the compatibility with  $O_c$ . This way, no set from an inferior level with a high  $I$  is better than a set from a superior level with low  $I$ .

In essence, the G-Indicator creates a partial order for the NSs and resolves the ties using a dispersion operator.

### 2.4 Computational Complexity

The most expensive part of the G-Indicator is the computation of the levels of dominance (see subsection 2.2). It implies comparing every NS with all the others for all the levels of dominance. In the worst case (when there is only one set per level) the number of comparisons is  $O(M^3)$  where  $M$  is the number of NSs. A comparison between two sets is  $O(dN^2)$  where  $N$  is the size of the sets and  $d$  is the number of dimensions, so the total complexity in the worst case is  $O(dM^3N^2)$ . For high dimensions this metric is efficient due to its linear complexity related to the number of objective functions.

## 3 Experiments and Results

In order to evaluate the performance of our approach, we designed some examples that consider various topologies of the Pareto front. We compare our results with those of the S-metric [2], a metric that stands out as one of the



most representative of the state of the art in literature. The S-metric measures convergence and dispersion at the same time. It assigns a real number to a set, the bigger the number, the better the set (at least for minimization problems). We present two cases here, the first one with five NSs and the second one with two NSs. We consider all the test cases as minimization problems. The result of the experiments are the following.

**Experiment 1.** This case is based on problem DTLZ1 [9]. In this problem, the Pareto front consists of all the points  $p$  with components  $p^k \geq 0$  and  $\sum_{i=1}^d p^i = 0.5$ . We created three cases, one in two dimensions (2d), another one in three dimensions (3d) and the last one in four dimensions (4d). In each case there are five NSs with different degrees of distribution. Figure 4 shows four of the five sets in 3d, the convergence for all of them is the same. The goal is to confirm whether the measures find the correct order. From the best to the worst, the order of the sets is A, B, C, D and E.

We can see in Table 1 that both measures identify correctly the level of distribution on the five sets. The values of the the S-metric get more similar as the dimension of the set increases, making it less clear which set is better than the others. The G-Indicator keeps a clear difference between its values.

**Table 1.** Results for G-Indicator and S-measure in Experiment 1

	2d		3d		4d	
Set	G-Indicator	S-measure	G-Indicator	S-measure	G-Indicator	S-measure
A	0.7354	0.4642	0.2311	0.7959	0.0354	0.9380
B	0.6237	0.4446	0.1685	0.7855	0.0267	0.9349
C	0.5934	0.4518	0.1617	0.7849	0.0266	0.9345
D	0.2413	0.2612	0.1486	0.7808	0.0203	0.9296
E	0.1470	0.2412	0.1290	0.7690	0.0130	0.9208

**Experiment 2.** The goal of this experiment is to compare the sensibility of the measures to the convexity of the Pareto front. Both NSs,  $A$  and  $B$  have the same extension, distribution and convergence, but  $A$  is on a non-convex zone while  $B$  is on a convex zone (Figure 5). We expect a similar value for both NSs.

It is clear, according to Table 2, that the S-metric has a bias towards the convex zones of the Pareto front thus it failed the test. Note that the G-Indicator is not affected by the convexity of the sets.

**Table 2.** Results for G-Indicator and S-measure in Experiment 2

Set	G-Indicator	S-measure
A	0.413250	0.298704
B	0.413250	0.440956

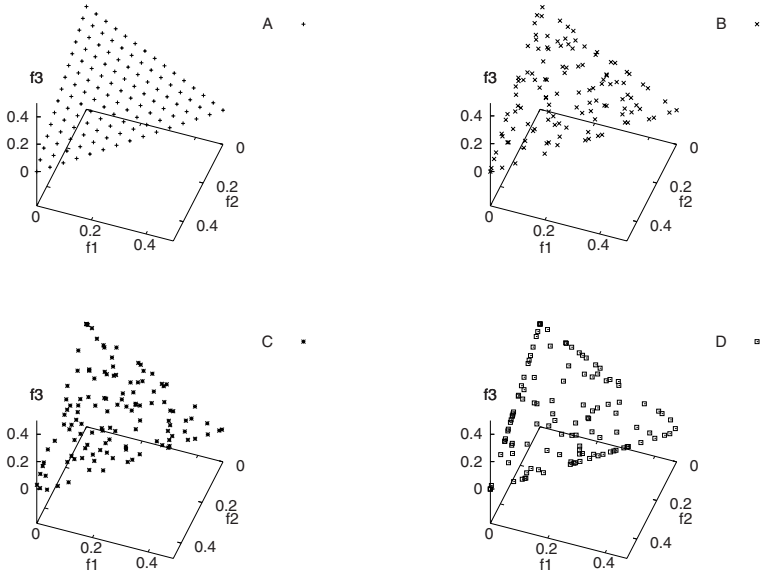


Fig. 4. The first four sets for Experiment 1

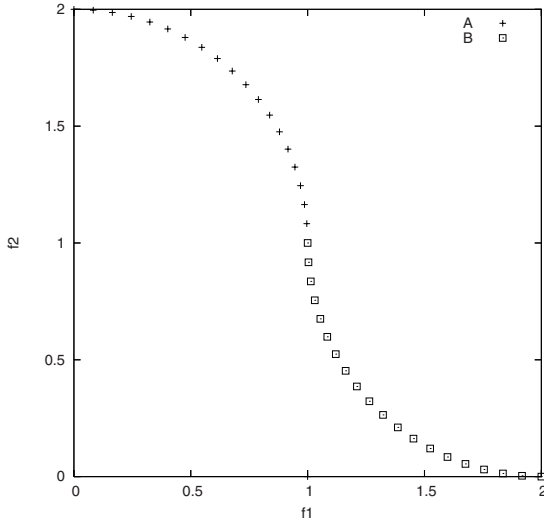


Fig. 5. Experiment 2

## 4 Conclusions

We present a new performance measure for the evaluation of NSs, the G-indicator. It is simple to understand and to implement, its complexity is low  $O(dM^3N^2)$  compared to other algorithms (the complexity of the S-metric is  $O(N^{d/2})$ ). It does not need any extra information, neither does it need further parameter tuning. It successfully combines convergence, distribution and extension in a single number, and its evaluations agree with intuition giving better scores to NSs with better convergence and extension. It is robust in misleading cases, like fronts with convex and non-convex zones. In order to evaluate the G-Indicator we created two test cases. In both cases our approach gave the correct answer showing a better performance than the S-measure.

## References

1. Leung, Y.-W., Wang, Y.-P.: U-Measure: a Quality Measure for Multiobjective Programming. *IEEE Transactions on Systems, Man, and Cybernetics Part A* 33(3), 337–343 (2003)
2. Zitzler, E.: Evolutionary Algorithms Multiobjective Optimization: Methods and Applications. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (1999)
3. Veldhuizen, D.A.: Multiobjective Evolution Algorithms: Classifications, Analyses, and New Innovations. PhD thesis, Department Electrical Computer Engineering, Graduate School Engineering, Force Institute Technology, Wright Patterson AFB, Ohio (1999)
4. Knowles, J., Corne, D.: On Metrics for Comparing Non-Dominated Sets, Congress on Evolutionary Computation. In: CEC 2002 (2002)
5. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: Nsga II. In: Proceedings of the Parallel Problem Solving from Nature VI Conference, Paris, France, 16–20 September, pp. 849–858 (2000)
6. Knowles, J., Corne, D.: The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation. In: Proceedings of the 1999 Congress on Evolutionary Computation, pp. 98–105. IEEE Service Center, New Jersey (1999)
7. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms. John Wiley and Sons, Chichester (2001)
8. Hansen, M.P., Jaszkiwicz, A.: Evaluating the Quality of Approximations to the Non-Dominated Set. Technical Report IMM-REP-1998-7, Technical University of Denmark (1998)
9. Deb, et al.: Scalable Test Problems for Evolutionary Multi-Objective Optimization. TIK-Technical Report No. 112 Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich Gloriastrasse 35., ETH-Zentrum, CH-8092, Zürich, Switzerland (2001)
10. Mehr, A.F., Azarm, S.: Minimal Sets of Quality Metrics. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 405–417. Springer, Heidelberg (2003)

# Approximating the $\epsilon$ -Efficient Set of an MOP with Stochastic Search Algorithms

Oliver Schütze<sup>1</sup>, Carlos A. Coello Coello<sup>1</sup>, and El-Ghazali Talbi<sup>2</sup>

<sup>1</sup> CINVESTAV-IPN, Computer Science Department  
México D.F. 07300, Mexico

{schuetze,ccoello}@cs.cinvestav.mx

<sup>2</sup> INRIA Futurs, LIFL, CNRS Bât M3, Cité Scientifique  
59655 Villeneuve d'Ascq, France  
talbi@lifl.fr

**Abstract.** In this paper we develop a framework for the approximation of the entire set of  $\epsilon$ -efficient solutions of a multi-objective optimization problem with stochastic search algorithms. For this, we propose the set of interest, investigate its topology and state a convergence result for a generic stochastic search algorithm toward this set of interest. Finally, we present some numerical results indicating the practicability of the novel approach.

## 1 Introduction

Since the notion of  $\epsilon$ -efficiency for multi-objective optimization problems (MOPs) has been introduced more than two decades ago ([6]), this concept has been studied and used by many researchers, e.g. to allow (or tolerate) nearly optimal solutions ([6], [13]), to approximate the set of optimal solutions ([9]), or in order to discretize this set ([5], [11]).  $\epsilon$ -efficient solutions or approximate solutions have also been used to tackle a variety of real world problems including portfolio selection problems ([14]), a location problem ([1]), or a minimal cost flow problem ([9]). The explicit computation of such approximate solutions has been addressed in several studies (e.g., [13], [1], [2]), in all of them scalarization techniques have been employed.

The scope of this paper is to develop a framework for the approximation of the set of  $\epsilon$ -efficient solutions (denote by  $E_\epsilon$ ) with stochastic search algorithms such as evolutionary multi-objective (EMO) algorithms. This calls for the design of a novel archiving strategy to store the ‘required’ solutions found by a stochastic search process (though the investigation of the set of interest will be the major part in this work). One interesting fact is that the solution set (the *Pareto set*) is included in  $E_\epsilon$  for all (small) values of  $\epsilon$ , and thus the resulting archiving strategy for EMO algorithms can be regarded as an alternative to existing methods for the approximation of this set (e.g. [3], [7], [4], [8]).

The remainder of this paper is organized as follows: in Section 2, we give the required background for the understanding of the sequel. In Section 3, we propose a set of interest, analyze its topology, and state a convergence result. We present numerical results on two examples in Section 4 and conclude in Section 5.

## 2 Background

In the following we consider continuous multi-objective optimization problems

$$\min_{x \in \mathbb{R}^n} \{F(x)\}, \tag{MOP}$$

where the function  $F$  is defined as the vector of the objective functions  $F : \mathbb{R}^n \rightarrow \mathbb{R}^k$ ,  $F(x) = (f_1(x), \dots, f_k(x))$ , and where each  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  is continuously differentiable. Later we will restrict the search to a compact set  $Q \subset \mathbb{R}^n$ , the reader may think of an  $n$ -dimensional box.

- Definition 1.** (a) Let  $v, w \in \mathbb{R}^k$ . Then the vector  $v$  is less than  $w$  ( $v <_p w$ ), if  $v_i < w_i$  for all  $i \in \{1, \dots, k\}$ . The relation  $\leq_p$  is defined analogously.  
 (b)  $y \in \mathbb{R}^n$  is dominated by a point  $x \in \mathbb{R}^n$  ( $x \prec y$ ) with respect to **(MOP)** if  $F(x) \leq_p F(y)$  and  $F(x) \neq F(y)$ , else  $y$  is called nondominated by  $x$ .  
 (c)  $x \in \mathbb{R}^n$  is called a Pareto point if there is no  $y \in \mathbb{R}^n$  which dominates  $x$ .  
 (d)  $x \in \mathbb{R}^n$  is weakly Pareto optimal if there does not exist another point  $y \in \mathbb{R}^n$  such that  $F(y) <_p F(x)$ .

We now define a weaker concept of dominance, called  $\epsilon$ -dominance, which is the basis of the approximation concept used in this study.

- Definition 2.** Let  $\epsilon = (\epsilon_1, \dots, \epsilon_k) \in \mathbb{R}_+^k$  and  $x, y \in \mathbb{R}^n$ .  $x$  is said to  $\epsilon$ -dominate  $y$  ( $x \prec_\epsilon y$ ) with respect to **(MOP)** if  $F(x) - \epsilon \leq_p F(y)$  and  $F(x) - \epsilon \neq F(y)$ .

**Theorem 1** (**[10]**). Let (MOP) be given and  $q : \mathbb{R}^n \rightarrow \mathbb{R}^n$  be defined by  $q(x) = \sum_{i=1}^k \hat{\alpha}_i \nabla f_i(x)$ , where  $\hat{\alpha}$  is a solution of

$$\min_{\alpha \in \mathbb{R}^k} \left\{ \left\| \sum_{i=1}^k \alpha_i \nabla f_i(x) \right\|_2^2; \alpha_i \geq 0, i = 1, \dots, k, \sum_{i=1}^k \alpha_i = 1 \right\}.$$

Then either  $q(x) = 0$  or  $-q(x)$  is a descent direction for all objective functions  $f_1, \dots, f_k$  in  $x$ . Hence, each  $x$  with  $q(x) = 0$  fulfills the first-order necessary condition for Pareto optimality.

In case  $q(x) \neq 0$  it obviously follows that  $q(x)$  is an ascent direction for all objectives. Next, we need the following distances between different sets.

**Definition 3.** Let  $u \in \mathbb{R}^n$  and  $A, B \subset \mathbb{R}^n$ . The semi-distance  $dist(\cdot, \cdot)$  and the Hausdorff distance  $d_H(\cdot, \cdot)$  are defined as follows:

- (a)  $dist(u, A) := \inf_{v \in A} \|u - v\|$   
 (b)  $dist(B, A) := \sup_{u \in B} dist(u, A)$   
 (c)  $d_H(A, B) := \max \{ dist(A, B), dist(B, A) \}$

Denote by  $\bar{A}$  the closure of a set  $A \in \mathbb{R}^n$ , by  $\overset{\circ}{A}$  its interior, and by  $\partial A = \bar{A} \setminus \overset{\circ}{A}$  the boundary of  $A$ .

Algorithm **[1]** gives a framework of a generic stochastic multi-objective optimization algorithm, which will be considered in this work. Here,  $Q \subset \mathbb{R}^n$  denotes the domain of the MOP,  $P_j$  the candidate set (or population) of the generation process at iteration step  $j$ , and  $A_j$  the corresponding archive.

---

**Algorithm 1.** Generic Stochastic Search Algorithm

---

- 1:  $P_0 \subset Q$  drawn at random
  - 2:  $A_0 = \text{ArchiveUpdate}(P_0, \emptyset)$
  - 3: **for**  $j = 0, 1, 2, \dots$  **do**
  - 4:      $P_{j+1} = \text{Generate}(P_j)$
  - 5:      $A_{j+1} = \text{ArchiveUpdate}(P_{j+1}, A_j)$
  - 6: **end for**
- 

### 3 The Archiving Strategy

In this section we define the set of interest, investigate the topology of this object, and finally state a convergence result.

**Definition 4.** Let  $\epsilon \in \mathbb{R}_+^k$  and  $x, y \in \mathbb{R}^n$ .  $x$  is said to  $-\epsilon$ -dominate  $y$  ( $x \prec_{-\epsilon} y$ ) with respect to (MOP) if  $F(x) + \epsilon \leq_p F(y)$  and  $F(x) + \epsilon \neq F(y)$ .

This definition is of course analogous to the ‘classical’  $\epsilon$ -dominance relation but with a value  $\tilde{\epsilon} \in \mathbb{R}_+^k$ . However, we highlight it here since it will be used frequently in this work. While the  $\epsilon$ -dominance is a weaker concept of dominance,  $-\epsilon$ -dominance is a stronger one.

**Definition 5.** A point  $x \in Q$  is called  $-\epsilon$  weak Pareto point if there exists no point  $y \in Q$  such that  $F(y) + \epsilon <_p F(x)$ .

Now we are able to define the set of interest. Ideally, we would like to obtain the ‘classical’ set

$$P_{Q,\epsilon}^c := \{x \in Q \mid \exists p \in P_Q : x \prec_{\epsilon} p\} \tag{1}$$

where  $P_Q$  denotes the Pareto set (i.e., the set of Pareto optimal solutions) of  $F|_Q$ . That is, every point  $x \in P_{Q,\epsilon}^c$  is ‘close’ to at least one efficient solution, measured in objective space. However, since this set is not easy to catch – note that the efficient solutions are used in the definition –, we will consider an enlarged set of interest (see Lemma 2):

**Definition 6.** Denote by  $P_{Q,\epsilon}$  the set of points in  $Q \subset \mathbb{R}^n$  which are not  $-\epsilon$ -dominated by any other point in  $Q$ , i.e.

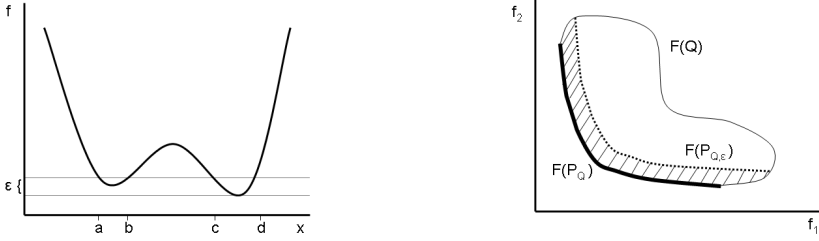
$$P_{Q,\epsilon} := \{x \in Q \mid \nexists y \in Q : y \prec_{-\epsilon} x\} \tag{2}$$

*Example 1.* (a) Figure 1 shows two examples for sets  $P_{Q,\epsilon}$ , one for the single-objective case (left), and one for  $k = 2$  (right). In the first case we have  $P_{Q,\epsilon} = [a, b] \cup [c, d]$ .

(b) Consider the MOP  $F : \mathbb{R} \rightarrow \mathbb{R}^2$ ,  $F(x) = ((x - 1)^2, (x + 1)^2)$ . For  $\epsilon = (1, 1)$  and  $Q$  sufficiently large, say  $Q = [-3, 3]$ , we obtain  $P_Q = [-1, 1]$  and  $P_{Q,\epsilon} = (-2, 2)$ . Note that the boundary of  $P_{Q,\epsilon}$ , i.e.  $\partial P_{Q,\epsilon} = \overline{P_{Q,\epsilon}} \setminus P_{Q,\epsilon}^\circ =$

---

<sup>1</sup>  $P_{Q,\epsilon}^c$  is closely related to set  $E^1$  considered in [13].  
<sup>2</sup>  $P_{Q,\epsilon}$  is closely related to set  $E^5$  considered in [13].



**Fig. 1.** Two different examples for sets  $P_{Q,\epsilon}$ . Left for  $k = 1$  and in parameter space, and right an example for  $k = 2$  in image space.

$[-2, 2] \setminus (2, 2) = \{-2, 2\}$ , is given by  $-\epsilon$  weak Pareto points which are not included in  $P_{Q,\epsilon}$  (see also Lemma 1): for  $x_1 = -2$  and  $x_2 = 2$  it is  $F(x_1) = (9, 1)$  and  $F(x_2) = (1, 9)$ . Since there exists no  $x \in Q$  with  $f_i(x) < 0, i = 1, 2$ , there is also no point  $x \in Q$  where all objectives are less than at  $x_1$  or  $x_2$ . Further, since  $F(-1) = (4, 0)$  and  $F(1) = (0, 4)$  there exist points which  $-\epsilon$ -dominate these points, and they are thus not included in  $P_{Q,\epsilon}$ .

- Lemma 1.** (a) Let  $Q \subset \mathbb{R}^n$  be compact. Under the following assumptions
- (A1) Let there be no weak Pareto point in  $Q \setminus P_Q$ , where  $P_Q$  denotes the set of Pareto points of  $F|_Q$ .
  - (A2) Let there be no  $-\epsilon$  weak Pareto point in  $Q \setminus \overline{P_{Q,\epsilon}}$ ,
  - (A3) Define  $\mathcal{B} := \{x \in Q \mid \exists y \in P_Q : F(y) + \epsilon = F(x)\}$ . Let  $\mathcal{B} \subset \overset{\circ}{Q}$  and  $q(x) \neq 0$  for all  $x \in \mathcal{B}$ , where  $q$  is as defined in Theorem [1](#), it holds:

$$\begin{aligned} \overline{P_{Q,\epsilon}} &= \{x \in Q \mid \nexists y \in Q : F(y) + \epsilon <_p F(x)\} \\ P_{Q,\epsilon}^\circ &= \{x \in Q \mid \nexists y \in Q : F(y) + \epsilon \leq_p F(x)\} \\ \partial P_{Q,\epsilon} &= \{x \in Q \mid \exists y_1 \in P_Q : F(y_1) + \epsilon \leq_p F(x) \wedge \nexists y_2 \in Q : F(y_2) + \epsilon <_p F(x)\} \end{aligned} \tag{3}$$

- (b) Let in addition to the assumptions made above be  $q(x) \neq 0 \forall x \in \partial P_{Q,\epsilon}$ . Then

$$\overset{\circ}{P_{Q,\epsilon}} = \overline{P_{Q,\epsilon}} \tag{4}$$

*Proof.* Define  $W := \{x \in Q \mid \nexists y \in Q : F(y) + \epsilon <_p F(x)\}$ . We show the equality  $\overline{P_{Q,\epsilon}} = W$  by mutual inclusion.  $W \subset \overline{P_{Q,\epsilon}}$  follows directly by assumption (A2). To see the other inclusion assume that there exists an  $x \in \overline{P_{Q,\epsilon}} \setminus W$ . Since  $x \notin W$  there exists a  $y \in Q$  such that  $F(y) + \epsilon <_p F(x)$ . Further, since  $F$  is continuous there exists further a neighborhood  $U$  of  $x$  such that  $F(y) + \epsilon <_p F(u), \forall u \in U$ . Thus,  $y$  is  $-\epsilon$ -dominating all  $u \in U$  (i.e.,  $U \cap P_{Q,\epsilon} = \emptyset$ ), a contradiction to the assumption that  $x \in \overline{P_{Q,\epsilon}}$ . Thus, we have  $\overline{P_{Q,\epsilon}} = W$  as claimed.

Next we show that the interior of  $P_{Q,\epsilon}$  is given by

$$I := \{x \in Q \mid \nexists y \in Q : F(y) + \epsilon \leq_p F(x)\}, \tag{5}$$

which we do again by mutual inclusion. To see that  $P_{Q,\epsilon}^\circ \subset I$  assume that there exists an  $x \in P_{Q,\epsilon}^\circ \setminus I$ . Since  $x \notin I$  we have

$$\exists y_1 \in Q : F(y_1) + \epsilon \leq_p F(x). \tag{6}$$

Since  $x \in P_{Q,\epsilon}^\circ$  there exists no  $y \in Q$  which  $-\epsilon$ -dominates  $x$ , and hence, equality holds in equation (6). Further, by assumption (A1) it follows that  $y_1$  must be in  $P_Q$ . Thus, we can reformulate (6) by

$$\exists y_1 \in P_Q : F(y_1) + \epsilon = F(x) \tag{7}$$

Since  $x \in P_{Q,\epsilon}^\circ$  there exists a neighborhood  $\tilde{U}$  of  $x$  such that  $\tilde{U} \subset P_{Q,\epsilon}^\circ$ . Further, since  $q(x) \neq 0$  by assumption (A1), there exists a point  $\tilde{x} \in \tilde{U}$  such that  $F(\tilde{x}) >_p F(x)$ . Combining this and (7) we obtain

$$F(y_1) + \epsilon = F(x) <_p F(\tilde{x}), \tag{8}$$

and thus  $y_1 \prec_{-\epsilon} \tilde{x} \in \tilde{U} \subset P_{Q,\epsilon}^\circ$ , which is a contradiction. It remains to show that  $I \subset P_{Q,\epsilon}^\circ$ : assume there exists an  $x \in I \setminus P_{Q,\epsilon}^\circ$ . Since  $x \notin P_{Q,\epsilon}^\circ$  there exists a sequence  $x_i \in Q \setminus P_{Q,\epsilon}^\circ, i \in \mathbb{N}$ , such that  $\lim_{i \rightarrow \infty} x_i = x$ . That is, there exists a sequence  $y_i \in Q$  such that  $y_i \prec_{-\epsilon} x_i$  for all  $i \in \mathbb{N}$ . Since all the  $y_i$  are inside  $Q$ , which is a bounded set, there exists a subsequence  $y_{i_j}, j \in \mathbb{N}$ , and an  $y \in Q$  such that  $\lim_{j \rightarrow \infty} y_{i_j} = y$  (Bolzano-Weierstrass). Since  $F(y_{i_j}) + \epsilon \leq_p F(x_{i_j}), \forall j \in \mathbb{N}$ , it follows for the limit points that also  $F(y) + \epsilon \leq_p F(x)$ , which is a contradiction to  $x \in I$ . Thus, we have  $P_{Q,\epsilon}^\circ = I$  as desired.

For the boundary we obtain

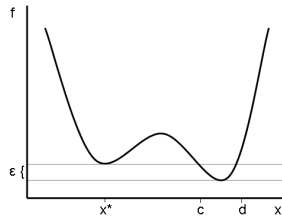
$$\begin{aligned} \partial P_{Q,\epsilon} &= \overline{P_{Q,\epsilon}^\circ} \setminus P_{Q,\epsilon}^\circ \\ &= \{x \in Q \mid \exists y_1 \in Q : F(y_1) + \epsilon \leq_p F(x) \text{ and } \nexists y_2 \in Q : F(y_2) + \epsilon <_p F(x)\} \end{aligned} \tag{9}$$

Since by (A1) the point  $y_1$  in (9) must be in  $P_Q$ , we obtain

$$\partial P_{Q,\epsilon} = \{x \in Q \mid \exists y_1 \in P_Q : F(y_1) + \epsilon \leq_p F(x) \text{ and } \nexists y_2 \in Q : F(y_2) + \epsilon <_p F(x)\} \tag{10}$$

It remains to show the second claim. It is  $\overline{P_{Q,\epsilon}^\circ} = P_{Q,\epsilon}^\circ \cup \partial P_{Q,\epsilon}$ . Assume that  $\overline{P_{Q,\epsilon}^\circ} \neq \overline{P_{Q,\epsilon}}$ , i.e., that there exists an  $x \in \partial P_{Q,\epsilon}$  and a neighborhood  $U$  of  $x$  such that  $U \cap P_{Q,\epsilon}^\circ = \emptyset$ . Since  $x \in \partial P_{Q,\epsilon}$  there exists a point  $y \in P_Q$  such that  $F(y) + \epsilon \leq_p F(x)$ . By assumption it is  $q(x) \neq 0$ , and thus there exists an  $\bar{x} \in U$  such that  $F(\bar{x}) <_p F(x)$ . Since  $\bar{x} \notin P_{Q,\epsilon}^\circ$  there exists an  $\bar{y} \in Q$  such that  $F(\bar{y}) + \epsilon \leq_p F(\bar{x}) <_p F(x)$ , which contradicts the assumption that  $x \in \partial P_{Q,\epsilon}$ . Thus, we have that the closure of the interior of  $P_{Q,\epsilon}$  is equal to its closure as claimed.





**Fig. 2.** Example of a set  $P_{Q,\epsilon}$  where the closure of its interior is not equal to its closure

*Remark 1.* (a) Note that in general  $P_{Q,\epsilon}$  is neither an open nor a closed set, and that  $\overline{P_{Q,\epsilon}}$  gets ‘completed’ by  $-\epsilon$  weak Pareto points (see also Example 1). (b) Since for  $x$  and  $y_1$  in equation (10) it must hold that there exists an index  $j \in \{1, \dots, k\}$  such that  $f_j(y_1) + \epsilon_j = f_j(x)$ . Thus, the boundary of  $P_{Q,\epsilon}$  can be characterized by the set of  $-\epsilon$  weak Pareto points which are bounded in objective space from  $P_Q$  by  $\epsilon$ .

The next example shows that the closure of the interior of  $P_{Q,\epsilon}$  does in general not have to be equal to its closure, which causes trouble to approximate  $\partial P_{Q,\epsilon}$  using stochastic search algorithms. However, the following Lemma shows that this is – despite for theoretical investigations – not problematic since  $\overset{\circ}{P_{Q,\epsilon}}$ , which can be approximated in any case, already contains all the interesting parts.

*Example 2.* Figure 2 shows an example which is a modification of the MOP in Example 1(a). We have  $P_{Q,\epsilon} = \{x^*\} \cup [c, d]$  and hence  $\overline{P_{Q,\epsilon}} = [c, d] \neq \overline{P_{Q,\epsilon}}$ . Note that here we have  $f'(x^*) = 0$ , and thus that (A3) is violated. The problem with the approximation of the entire set  $P_{Q,\epsilon}$  in this case is the following: assume that  $\text{argmin} f$  is already a member of the archive, then every candidate solution near  $x^*$  will be rejected by all further archives. Thus, the entire set  $P_{Q,\epsilon}$  can only be approximated if  $x^*$  is a member of a population  $P_i, i \in \mathbb{N}$ , and the probability for this event is zero. Such problems do not occur for points in  $\overset{\circ}{P_{Q,\epsilon}}$  (see proof of Theorem 2).

**Lemma 2.**  $P_{Q,\epsilon}^c \subset \overset{\circ}{P_{Q,\epsilon}}$

*Proof.* Assume there exists an  $x \in P_{Q,\epsilon}^c \setminus \overset{\circ}{P_{Q,\epsilon}}$ . Since  $x \in P_{Q,\epsilon}^c$  there exists a Pareto optimal point  $p \in P_Q$  with  $p \prec_\epsilon x$ . Further, since  $x \notin \overset{\circ}{P_{Q,\epsilon}}$  there exists an  $y \in Q$  such that  $F(y) + \epsilon \leq_p F(x)$ . Combining both we obtain

$$\begin{aligned} F(y) \leq_p F(x) - \epsilon \leq F(p), \quad \text{and} \\ \exists j \in \{1, \dots, k\} : f_j(y) \leq f_j(x) - \epsilon < f_j(p) \quad (\Rightarrow F(y) \neq F(p)), \end{aligned} \tag{11}$$

which means that  $y \prec p$ , a contradiction to  $p \in P_Q$ , and we are done.

Having analyzed the topology of  $P_{Q,\epsilon}$  we are now in the position to state the following result. The archiving strategy is simply the one which keeps all obtained points which are not  $-\epsilon$ -dominated by any other test point.

**Theorem 2.** *Let an MOP  $F : \mathbb{R}^n \rightarrow \mathbb{R}^k$  be given, where  $F$  is continuous, let  $Q \subset \mathbb{R}^n$  be a compact set and  $\epsilon \in \mathbb{R}_+^k$ . Further let*

$$\forall x \in Q \text{ and } \forall \delta > 0 : P(\exists l \in \mathbb{N} : P_l \cap B_\delta(x) \cap Q \neq \emptyset) = 1 \quad (12)$$

*Then, under the assumptions made in Lemma 1, an application of Algorithm 1, where*

$$\text{ArchiveUpdate}_{P_{Q,\epsilon}}(P, A) := \{x \in P \cup A : y \not\prec_{-\epsilon} x \ \forall y \in P \cup A\}, \quad (13)$$

*is used to update the archive, leads to a sequence of archives  $A_l, l \in \mathbb{N}$ , with*

$$\lim_{l \rightarrow \infty} d_H(P_{Q,\epsilon}, A_l) = 0, \quad \text{with probability one.} \quad (14)$$

*Proof.* Since  $\text{dist}(A, B) = \text{dist}(\overline{A}, B)$  for all sets  $A, B \subset \mathbb{R}^n$  and since  $\overset{\circ}{P}_{Q,\epsilon} = \overline{\overset{\circ}{P}_{Q,\epsilon}}$  (see Lemma 1), it is sufficient to show that the Hausdorff distance between  $A_l$  and  $\overset{\circ}{P}_{Q,\epsilon}$  vanishes in the limit with probability one.

First we show that  $\text{dist}(A_l, \overset{\circ}{P}_{Q,\epsilon}) \rightarrow 0$  with probability one for  $l \rightarrow \infty$ . It is

$$\text{dist}(A_l, \overset{\circ}{P}_{Q,\epsilon}) = \max_{a \in A_l} \inf_{p \in \overset{\circ}{P}_{Q,\epsilon}} \|a - p\|.$$

We have to show that every  $x \in Q \setminus \overline{\overset{\circ}{P}_{Q,\epsilon}}$  will be discarded (if added before) from the archive after finitely many steps, and that this point will never be added further on.

Let  $x \in Q \setminus \overline{\overset{\circ}{P}_{Q,\epsilon}}$ . Since  $x$  is by assumption (A2) not a  $-\epsilon$ -weak Pareto point, there exists a point  $p \in P_{Q,\epsilon}$  such that  $F(p) + \epsilon <_p F(x)$ . Further, since  $F$  is continuous there exists a neighborhood  $U$  of  $x$  such that

$$F(p) + \epsilon <_p F(u), \quad \forall u \in U. \quad (15)$$

By (12) it follows that there exists with probability one a number  $l_0 \in \mathbb{N}$  such that there exists a point  $x_{l_0} \in P_{l_0} \cap U \cap Q$ . Thus, by construction of  $\text{ArchiveUpdate}_{P_{Q,\epsilon}}$ , the point  $x$  will be discarded from the archive if it is a member of  $A_{l_0}$ , and never be added to the archive further on.

Now we consider the limit behavior of  $\text{dist}(\overset{\circ}{P}_{P,\epsilon}, A_l)$ . It is

$$\text{dist}(\overset{\circ}{P}_{Q,\epsilon}, A_l) = \sup_{p \in \overset{\circ}{P}_{Q,\epsilon}} \min_{a \in A_l} \|p - a\|.$$

Let  $\bar{p} \in \overset{\circ}{P}_{Q,\epsilon}$ . For  $i \in \mathbb{N}$  there exists by (12) a number  $l_i$  and a point  $p_i \in P_{l_i} \cap B_{1/i}(\bar{p}) \cap Q$ , where  $B_\delta(p)$  denotes the open ball with center  $p$  and radius  $\delta \in \mathbb{R}_+$ . Since  $\lim_{i \rightarrow \infty} p_i = \bar{p}$  and since  $\bar{p} \in \overset{\circ}{P}_{Q,\epsilon}$  there exists an  $i_0 \in \mathbb{N}$  such that  $p_i \in \overset{\circ}{P}_{Q,\epsilon}$  for all  $i \geq i_0$ . By construction of  $\text{ArchiveUpdate}_{P_{Q,\epsilon}}$ , all the points  $p_i, i \geq i_0$ , will be added to the archive (and never discarded further on). Thus, we have  $\text{dist}(\bar{p}, A_l) \rightarrow 0$  for  $l \rightarrow \infty$  as desired, which completes the proof.

*Remark 2.* In order to obtain a ‘complete’ convergence result we have postulated some (mild) assumptions in order to guarantee that  $\overline{P_{Q,\epsilon}^\circ} = \overline{P_{Q,\epsilon}}$ , which is in fact an important topological property needed for the proof. However, if we drop the assumptions we can still expect that the interior of  $P_{Q,\epsilon}$  – the ‘interesting’ part (see Lemma 2) – will be approximated in the limit. To be more precise, regardless of assumptions (A1)–(A3) it holds in the above theorem that

$$\lim_{l \rightarrow \infty} \overline{\text{dist}(P_{Q,\epsilon}, A_l)} = 0, \quad \text{with probability one.}$$

## 4 Numerical Results

Here we demonstrate the practicability of the novel archiver on two examples. For this, we compare *ArchiveUpdate* $P_{Q,\epsilon}$  against the ‘classical’ archiving strategy which stores all nondominated solutions obtained during the search procedure (*ArchiveUpdateND*). To obtain a fair comparison of the two archivers we have decided to take a random search operator for the generation process (the same sequence of points for all settings).

### 4.1 Example 1

First we consider the MOP suggested by Tanaka (12):

$$F : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad F(x_1, x_2) = (x_1, x_2) \tag{16}$$

where

$$\begin{aligned} C_1(x) &= x_1^2 + x_2^2 - 1 - 0.1 \cos(16 \arctan(x_1/x_2)) \geq 0 \\ C_2(x) &= (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5 \end{aligned}$$

Figure 3 shows two comparisons for  $N = 10,000$  and  $N = 100,000$  points within  $Q = [0, \pi]^2$  as domain 3, indicating that the method is capable of finding all approximate solutions.

### 4.2 Example 2

Finally, we consider the following MOP proposed in 8:

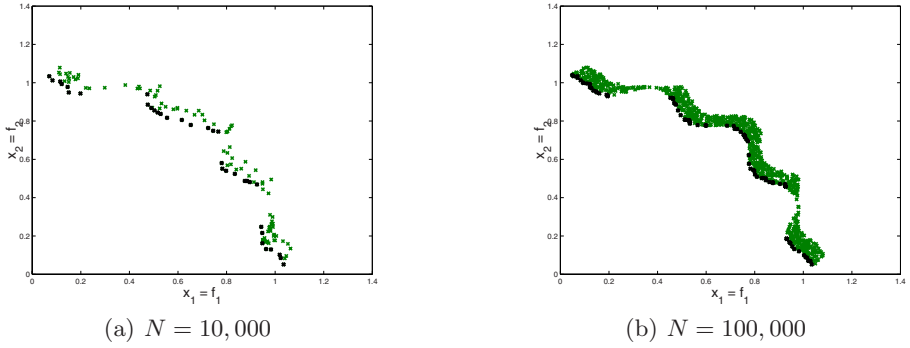
$$\begin{aligned} F : \mathbb{R}^2 &\rightarrow \mathbb{R}^2 \\ F(x_1, x_2) &= \left( \begin{aligned} (x_1 - t_1(c + 2a) + a)^2 + (x_2 - t_2b)^2 \\ (x_1 - t_1(c + 2a) - a)^2 + (x_2 - t_2b)^2 \end{aligned} \right), \end{aligned} \tag{17}$$

where

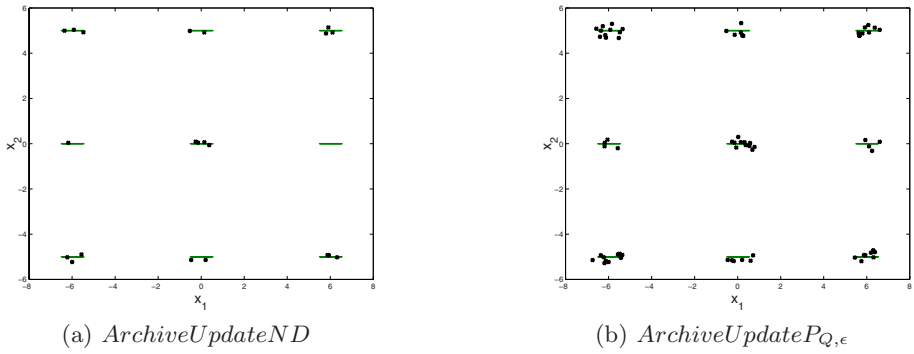
$$t_1 = \text{sgn}(x_1) \min \left( \left\lceil \frac{|x_1| - a - c/2}{2a + c} \right\rceil, 1 \right), \quad t_2 = \text{sgn}(x_2) \min \left( \left\lceil \frac{|x_2| - b/2}{b} \right\rceil, 1 \right).$$

---

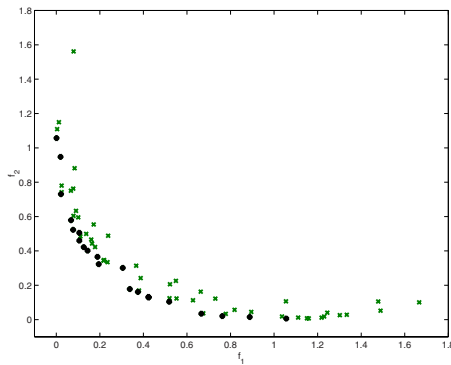
<sup>3</sup> To fit into our framework, we consider in fact the (compact) domain  $Q' := [0, \pi]^2 \cap \{x \in \mathbb{R}^n : C_1(x) \geq 0 \text{ and } C_2(x) \leq 0.5\}$ .



**Fig. 3.** Numerical result for MOP (16) using  $\epsilon = (0.1, 0.1)$



**Fig. 4.** Numerical result for MOP (16) in parameter space



**Fig. 5.** Comparison of the result of both archivers in objective space

The Pareto set consists of nine connected components of length  $a$  with identical images. We have chosen the values  $a = 0.5$ ,  $b = c = 5$ ,  $\epsilon = (0.1, 0.1)$ , the domain  $Q = [-20, 20]^2$ , and  $N = 10,000$  randomly chosen points within  $Q$ . Figures 4 and 5 display two typical results in parameter space and image space respectively. Seemingly, the approximation quality of the Pareto set obtained by the limit set of *ArchiveUpdate* $P_{Q,\epsilon}$  is better than by the one obtained by *ArchiveUpdate* $ND$ , measured in the Hausdorff sense. This example should indicate that it can be advantageous to store more than just non-dominated points in the archive, even when ‘only’ aiming for the efficient set.

## 5 Conclusion and Future Work

We have proposed and investigated a novel archiving strategy for stochastic search algorithms which allows – under mild assumptions on the generation process – to approximate the set  $P_{Q,\epsilon}$  which contains all  $\epsilon$ -efficient solutions within a compact domain  $Q$ . We have proven the convergence of the algorithm toward this set in the probabilistic sense, and have given two examples indicating the usefulness of the approach.

Since the set of approximate solutions forms an  $n$ -dimensional object, a suitable finite size representation of  $P_{Q,\epsilon}$  and the related archiving strategy are of major interest for further investigations.

## Acknowledgements

The authors would like to thank the referees whose comments were helpful to refine the scope of the paper. The second author gratefully acknowledges support from CONACyT project no. 45683-Y.

## References

1. Blanquero, R., Carrizosa, E.: A. d.c. biobjective location model. *Journal of Global Optimization* 23(2), 569–580 (2002)
2. Engau, A., Wiecek, M.M.: Generating epsilon-efficient solutions in multiobjective programming. *European Journal of Operational Research* 177(3), 1566–1579 (2007)
3. Hanne, T.: On the convergence of multiobjective evolutionary algorithms. *European Journal Of Operational Research* 117(3), 553–564 (1999)
4. Knowles, J., Corne, D.: Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation* 7(2), 100–116 (2003)
5. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation* 10(3), 263–282 (2002)
6. Loridan, P.:  $\epsilon$ -solutions in vector minimization problems. *Journal of Optimization, Theory and Application* 42, 265–276 (1984)

7. Rudolph, G., Agapie, A.: On a multi-objective evolutionary algorithm and its convergence to the Pareto set. In: CEC 2000. Congress on Evolutionary Computation, pp. 1010–1016 (2000)
8. Rudolph, G., Naujoks, B., Preuss, M.: Capabilities of EMOA to detect and preserve equivalent Pareto subsets. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 36–50. Springer, Heidelberg (2007)
9. Ruhe, G., Fruhwirt, B.:  $\epsilon$ -optimality for bicriteria programs and its application to minimum cost flows. *Computing* 44, 21–34 (1990)
10. Schaeffler, S., Schultz, R., Weinzierl, K.: Stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization Theory and Applications* 114(1), 209–222 (2002)
11. Schütze, O., Laumanns, M., Tantar, E., Coello Coello, C.A., Talbi, E.-G.: Convergence of stochastic search algorithms to gap-free Pareto front approximations. In: GECCO 2007. Proceedings of the Genetic and Evolutionary Computation Conference (2007)
12. Tanaka, M.: GA-based decision support system for multi-criteria optimization. In: Proceedings of International Conference on Systems, Man and Cybernetics, pp. 1556–1561 (1995)
13. White, D.J.: Epsilon efficiency. *Journal of Optimization Theory and Applications* 49(2), 319–337 (1986)
14. White, D.J.: Epsilon-dominating solutions in mean-variance portfolio analysis. *European Journal of Operational Research* 105(3), 457–466 (1998)

# A Multicriterion SDSS for the Space Process Control: Towards a Hybrid Approach

Hamdadou Djamil<sup>1</sup> and Bouamrane Karim<sup>2</sup>

<sup>1</sup> Department of computer science, Faculty of Sciences, University of Oran Es-Senia  
dzhammadoud@yahoo.fr

<sup>2</sup> Department of computer science, Faculty of Sciences, University of Oran Es-Senia  
kbouamranedz@yahoo.fr

**Abstract.** Multicriterion classification methods traditionally employed in the spatial decision-making penalize the complex phenomena of interaction between the criteria. Indeed, the most classical procedure in the multicriterion evaluation consists in considering a simple weighted arithmetical average to incorporate information characterizing decision maker's preferences on the set of criteria. However, in reality, the criteria interact (correlation, interchangeability, complementarity, ...) and the preferential independence hypothesis is rarely checked. The ultimate goal of this study is to optimize the quality of decision in the land management context. Therefore, a decision support model is designed to meet this objective and to claim with an extensible, generic and deterministic model based on the axiomatic of decision strategies authorizing the interactions between criteria. We define a new approach replacing the additivity property in the performance aggregation phase by a more reliable property: the growth using non-additive discriminant operators resulting from the fuzzy theory: Choquet's Integral. Also, the suggested model allows the professionals to carry out diagnostics and proposes adapted actions by modelling the multi-actor negotiation and participation using multi-agents systems.

**Keywords:** Choquets Integral, Linear Programming, Multi-Actor Negotiation, Multicriterion Decision Making, Territory Planning.

## 1 Introduction

Because the social aspiration to the administrative decision transparency particularly in the environment field becomes a stake, the decision-making process changes from a traditional downward approach towards a new logic where the decisional power is redistributed. In order to interpret and integrate informations on the environmental quality, the environmentalist needs answers to the following questions:

*Question 1.* Which decision-making procedure is necessary to be adapted to the environment field?

*Question 2.* How to ensure the decision makers' negotiation and participation?

The essence of this paper is to propose a multicriterion **Spatial Decision Support System (SDSS)** devoted to help the deciders to better analyze the territorial context in the presence of several actors. Our purpose consists on a strategy for integrating "Geographical Information Systems" GIS, "Multicriterion Analysis Methods" and "Multi Agents Systems". The main benefit of this strategy is to optimize the aggregation phase by considering interactive aspects between the criteria by using the Choquets integral as a discriminant function. The rest of this paper is organized as follows: once the context of our study is specified, section 2, briefly presents reviews of related works. Section 3 clarifies the limits of the additive models justifying the opportunity of exploiting the non-additive ones in the multicriterion aggregation. In sections 4, we introduce the concepts of fuzzy measurements and Choquet's integral. Section 5 describes the identification model of the fuzzy measurement and outlines the algorithms developed over this model. In section 6, we describe in detail the decisional model. The suggested approach is accompanied by an experimental study described in section 7 and focusing on the various phases of the SDSS. Finally, section 8 concludes the paper by summarizing our work and providing a preview of future perspectives.

## 2 Related Works

Several authors have already showed the adequacy of the innovator association of GIS to the multicriterion analysis methods for the service of the decision-making (Ranking, Choice, Sorting and Description) in Territory Planning (TP). The demand is increasing in the environmental and urban development sectors since the price objectives are no longer the only ones to justify a decision. In [4], the author has approached the best adapted site for a factory of carpet manufacturing. [3] has integrated environmental criteria into the traditional technical criteria in order to choose the best scenario of equipment in power lines in Quebec. In [12], many applications of multicriterion methods concerning the environment management have been described and in [8], MEDUSAT is proposed for the treatment of localization problems. However, the authors do not attach importance to the complex phenomena of interactive criteria such as correlation, complementarity, interchangeability and preferential dependence into the aggregation phase, primarily into the weight determination process. In order to set up a deterministic multicriterion model characterizing the decision strategies in consideration of the interactions between criteria and on the basis of a participative and interactive procedure, our idea is to propose an original model based jointly on: Capacity theory, operational research, spatial analysis and Multi-Agents Systems. Moreover, the present research aims to propose a coherent SDSS able to support two problems in TP: the first relates to the search of a surface better satisfying certain criteria (the punctual management), the second consists in the geographical chart segmentation in areas (realizing the land use plan).



### 3 Additive Models: Criticism

In the multicriterion decision-making procedure, when the preferential independence between criteria is supposed, it is frequent to consider the classical additive model within the phase of performance aggregation. The weighted arithmetical average (WAA) is the most used aggregation operator:

$$M_w(x) = \sum_{i=1}^n \omega_i \cdot x_i / x \in [0, 1]^n \text{ and } \sum_{i=1}^n \omega_i = 1, \omega_i \geq 0, \forall i \in N \quad (1)$$

$N$ : indicates the set of  $n$  indices relative to the identified criteria and  $\omega_i$  the weight<sup>1</sup> (or importance) of the criterion  $i$ .

It is acquired that the additivity function is not always a required property in real situations, particularly in the presence of human reasoning where the preferential independence hypothesis is seldom checked. Indeed, WAA is unable to model interactive criteria. Also, it:

- Gums the possible conflicting character of the criteria;
- Eliminates from the Pareto-optimal alternatives<sup>2</sup> which can be interesting;
- Can favour the extreme alternatives;

### 4 Non-additive Models

In the multicriterion aggregation, we have recourse to these models when the separability property is not checked. The latter were proposed by Sugeno [13] to generalize additive measurements and to express synergies between criteria. Among the most used non-additive aggregation functions, we cite: the non-additive integrals with regard to a capacity, the most known are Choquet’s integral and Sugeno’s integral [9].

#### 4.1 Fuzzy Measurements

**Definition 1.** A fuzzy measurement  $\nu$  on  $N$  is a monotonous overall function  $\nu : 2^N \rightarrow [0, 1]$  i.e  $\nu(S) \leq \nu(T)$  each time  $S \subseteq T (S, T \subseteq N)$  and checks the limiting conditions  $\nu(\{\}) = 0$  and  $\nu(N) = 1$ .

$F_N$  denotes the set of all the fuzzy measurements on  $N$ . For any  $S \subseteq N, \nu(S)$  can be interpreted as the weight (or the importance coefficient) of the criteria combination  $S$ . Better still, it is its importance or its capacity to make alone the decision (without intervention of the other criteria) [9].

---

<sup>1</sup> We will always suppose that the weights are numerical values definite on a cardinal scale.

<sup>2</sup> An alternative  $a$  is Pareto-optimal or effective if it is not dominated by any other one. It cannot be improved with regard to a criterion without deteriorating it for another one.

### 4.2 Choquet’s Integral, Definition and Intuitive Approach

The concept of Choquet’s integral has been initially introduced in the capacity theory [1]. Its use as a fuzzy integral compared to a fuzzy measurement has been then proposed by Murofushi and Sugeno [9]. This operator can be seen as the simplest means to extend the decision maker’s reasoning on binary alternatives. In this section, we propose an axiomatic characterization of this operator.

**Definition 2.** Let  $\nu \in F_N$ , Choquet’s integral of the function  $x : N \rightarrow IR$  compared to  $\nu$  is defined by:

$$C_\nu(x) = \sum_{i=1}^n x_{(i)}[\nu(A_{(i)}) - \nu(A_{(i-1)})] \tag{2}$$

Where  $(.)$  indicates a permutation of  $N$  such that  $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}$ .  
 Moreover  $A_{(i)} = \{(i), k, (n)\}$  and  $A_{(n-1)} = \{(n)\}$

*Intuitive Approach and Properties.* Given that  $\nu \in F_N$ , we search for an aggregation operator  $M_\nu : IR^n \rightarrow IR$  which generalises the arithmetical weighted average and is identified with the latter as soon as  $\nu$  is additive.

As an aggregation operator, Choquet’s integral is a monotonous increasing function, defined of  $[0, 1]^N$  in  $[0, 1]$  limited by two values  $C_\nu(0, \dots, 0) = 0$  and  $C_\nu(1, \dots, 1) = 1$  and satisfying particularly remarkable properties of *continuity*, *idempotence* and *decomposability* [9].

### 4.3 K-Order Additive Fuzzy Measurement

In the decisional problems including  $n$  criteria, to be able to consider interaction among the criteria in the decision maker’s preference modeling, we need to define  $2^n$  coefficients representing the fuzzy measurement  $\nu$ , where each coefficient corresponds to the weight of a subset of  $N$ . However, the decision maker cannot provide the totality of information allowing to identify these coefficients. In the best cases, he can guess the importance of each criterion or each pair of criterion.

In order to avoid this problem, Grabish [5] has proposed the concept of  $k$ -order additive fuzzy measurement.

**Definition 3.** A fuzzy measurement  $\nu$  defined on  $N$  is additive of order  $K$ , if the boolean function corresponding to this measurement is a multilinear polynomial of order  $k$ . Choquet’s integral compared to an additive measurement of order 2 (a model of order 2 of Choquet’s integral) becomes:

$$C_\nu(g) = \sum_{i \in N} a(i)g_{(i)} + \sum_{\{i,j\} \subset N} a(ij)[g_{(i)} \wedge g_{(j)}]; \forall g \in R^n \tag{3}$$

This Choquet’s integral model of order 2 allows modeling the interaction among the criteria by using only  $n + C_n^2 = n(n + 1)/2$  coefficients to define the fuzzy measurement. We will use this model for the determination of the weights in the developed sorting approaches.

## 5 The Proposed Model

In this work, we develop a multicriterion model referring to the use of discrete Choquet’s integral according to a fuzzy measurement. The advantage of this model is its power to consider the interaction phenomena among the criteria in the preference aggregation [7]. The proposed sorting procedure thus can be regarded as an extension of ELECTRE Tri [11].

### 5.1 Proposition of an Identification Model of the Fuzzy Measurement

Marichal model [9] for identification of a fuzzy measurement is based on the definition of a partial quasi-order in the set of actions A, a partial preorder in the set of criteria F and the semantic considerations around these criteria. The latter concerns: the importance of the criteria and their interactions.

Information given by this model is ”Poor”. However, the fact of defining a partial arrangement on F according to the importance criterion  $\omega(j)/j \in F$  do not identify precisely the criteria importance coefficients  $\omega_j$ . Consequently, to make this model more deterministic as for the determination of the criteria importance coefficients and the interaction indices, we propose to consider, moreover, the limits of  $\omega_j$  ( by the intervals of the form  $[\omega_j^-, \omega_j^+] / j \in F$ ). In what follows, we will analyze the problem of the identification of these coefficients  $\omega_j/j \in F$  and  $\omega_{ij}/i, j \in F$ .

Formally, the input data of the suggested model are:

1. The set of actions to sorting  $A = \{a_1, a_2, \dots, a_m\}$ ;
2. The coherent criterion family  $F = \{g_1, g_2, \dots, g_n\}$ ;
3. The table of performances  $(g_j(a_i))$  (the decision maker associates with each action  $a_i$  the performance  $(g_j(a_i))$  of each criterion  $g_j$ ) and samples of assignment;
4. A partial quasi-order in A  $\geq_A$  (a partial arrangement of the actions according to their total performances);
5. A partial preorder  $\geq_F$  in F (a partial arrangement of the criterion according to their importance coefficients);
6. A partial preorder  $\geq_P$  in the set of criteria pairs P (a partial arrangement of the criteria pairs according to their interaction indices);
7. The sign of certain interaction indices  $\omega_{ij} > 0, = 0$  or  $< 0$  representing respectively a positive synergy, an independence or a redundancy between the criteria i and j;
8. The limits of  $\omega_j$  ( $[\omega_j^-, \omega_j^+] / j \in F$ );

All these data are modeled in terms of linear equations (or inequations) according to the fuzzy measurement  $\nu$ . Thus, the determination of  $\omega_j$  and  $\omega_{ij}$  consists in solving the appropriate constraint satisfaction problem.

## 5.2 Proposal of an Ordinal Sorting Algorithm Founded on Choquet's Integral

The ordinal sorting strategy implies a synthesis outclassing procedure which rests on a preferences model accepting the situations of incomparability between the actions . It allows assigning the action  $a_i \in A$  to a category  $C_h$ . The categories are ordered  $C = \{C_1, \dots, C_p\}$  and characterized by two limiting reference actions sequence (the lower limit and the upper limit).

The multicriterion algorithm introduced into this section is founded on Choquet's integral and is carried out through two phases:the categories modeling and the conjunctive assignment procedure.

The determination of  $\omega_j$  and  $\omega_{ij}$  is ensured by solving the following linear program **(P1)**:

$$(P1) \left\{ \begin{array}{l} \text{Max } z = \epsilon \\ \omega_i - \omega_j \geq \epsilon ; \quad i \succ_F j \quad [1] \\ \omega_i = \omega_j ; \quad i \approx_F j \quad [1] \\ \omega_{ij} - \omega_{kl} \geq \epsilon ; \quad ij \succ_p kl \quad [2] \\ \omega_{ij} = \omega_{kl} ; \quad ij \approx_p kl \quad [2] \\ \omega_{ij} \geq \epsilon \text{ (resp } \leq \epsilon) \text{ If } \omega_{ij} > 0 \text{ resp } (\omega_{ij} < 0) \text{ Else } (\omega_{ij} = 0) \quad [3] \\ \sum_i \omega_i + \sum_{i,j} \omega_{ij} = 1 ; \quad \forall i, j \in F; \quad [4] \\ \omega_i \geq 0 ; \quad \forall i \in F \quad [4] \\ \omega_i + \sum_j \omega_{ij} \geq 0 ; \quad \forall i \in F, \quad \forall j \in T, T \subseteq F - \{i\} \quad [4] \\ \omega_j^- \leq \omega_j \leq \omega_j^+ ; \quad \forall j \in F \quad [5] \\ \sum_i \omega_i c_i(a_r, b_h) + \sum_{i,j} \omega_{ij} [c_i(a_r, b_h) \wedge c_j(a_r, b_h)] < \lambda; \forall a_r \in A_1, i, j \in F \quad [6] \\ \sum_i \omega_i c_i(a_r, b_{h-1}) + \sum_{i,j} \omega_{ij} [c_i(a_r, b_{h-1}) \wedge c_j(a_r, b_{h-1})] \geq \lambda \quad [6] \end{array} \right.$$

- [1]: An arrangement of the criteria according to their importance coefficients.
- [2]: An arrangement of the criteria pairs according to their interaction indices.
- [3]: The sign of certain interaction indices.
- [4]: The conditions of the monotonicity limits.
- [5]: The limits of  $\omega_j$ .
- [6]:  $A_1$  an assignment sample r:the number of central reference actions in the category  $C_h$ .

### Ordinal Sorting Algorithm (*Ord-Choquet*)

$A, F, C$ : Respectively the sets of actions, criteria and categories;

$B = \{b_1, b_2, \dots, b_n\}$ : The set of the limiting reference actions;

$q_j(b_h), p_j(b_h), \nu_j(b_h)$ : Respectively, the indifference, the preference and the veto thresholds;

$\lambda$ : The cut value; this parameter ensures that the action compared with a category profiles satisfies the principle of majority;

**For** i= 1 **to** m **Do**

**For** h= 0 **to** p **Do**

**For** j= 1 **to** n **Do**

      Calculate the partial agreement index  $c_j(a_i, b_h)$ ;

$$c_j(a_i, b_h) \leftarrow \frac{p_j(b_h) - \min\{g_j(b_h) - g_j(a_i), p_j(b_h)\}}{p_j(b_h) - \min\{g_j(b_h) - g_j(a_i), q_j(b_h)\}}$$

      Calculate the partial disagreement index  $d_j(a_i, b_h)$ ;

$$d_j(a_i, b_h) \leftarrow \min \left\{ 1, \max \left\{ 0, \frac{g_j(b_h) - g_j(a_i) - p_j(b_h)}{\nu_j(b_h) - p_j(b_h)} \right\} \right\};$$

**EndDo;**

Calculate the total agreement index  $C(a_i, b_h)$ ;

$$C(a_i, b_h) \leftarrow \sum_{j \in F} \omega_j \cdot c_j(a_i, b_h) + \sum_{j, k \in F} \omega_{jk} \cdot (c_j(a_i, b_h) \wedge c_k(a_i, b_h));$$

**If**  $d_j(a_i, b_h) > C(a_i, b_h)$  **then**

Calculate the credibility index  $\sigma(a_i, b_h)$ ;

$$\sigma(a_i, b_h) \leftarrow C(a_i, b_h) \cdot \prod_{j \in F} \frac{1 - d_j(a_i, b_h)}{1 - C(a_i, b_h)};$$

**EndIf;**

**EndDo;**

**EndDo;**

**Conjunctive assignment procedure**

**For**  $i = 1$  **to**  $m$  **Do**

**For**  $h = p$  **downto**  $0$  **Do**

**If**  $\sigma(a_i, b_h) \geq \lambda$  **then** break;

**EndDo;**

Assign  $a_i$  to  $C_{h+1}$ ;

**EndDo;**

### 5.3 Proposal of a Nominal Sorting Algorithm Founded on Choquet's Integral

In this section, we deal with the nominal sorting procedure by considering the interactive aspect between the criteria. This strategy aims at helping the decision maker to choose the most possible classes to the assignment of an action  $a$ .

The determination of  $\omega_j / j \in F$  and  $\omega_{ij} / (i, j) \in F$  is ensured by solving the linear program (P2). P2 is the same to P1 at the difference of the two last inequations (samples of assignment) which are replaced by the following inequation:

$$\sum_{i \in F} \omega_i c_i(a_r, b_p^h) + \sum_{\{i, j \in F\}} \omega_{ij} [c_i(a_r, b_p^h) \wedge c_j(a_r, b_p^h)] \geq \lambda; \forall a_r \in A_1 \quad (4)$$

## 6 The Decionnal Model Description

Our decision-making aid approach out of TP is voluntarist, but not interventionist, decentralized, flexible, opened and participative. Also, in our decisional activity, spatial problems of a, *Multi Scale, Multi Actor, Multi Objective and Multi Criterion* nature are raised.

*The Territory Model.* The couple (GIS, simulation model) constitutes a model to describe the territory in great details, it is the support of the spatial analysis procedures. When the decision makers identify the actions and the criteria, these procedures allow attributing relatively, to the various actions, a value (performance) on each criterion. The set of the actions and their performances constitutes the evaluation matrix (or the table of performances). In the proposed model, this matrix is managed by the GIS (MapInfo).

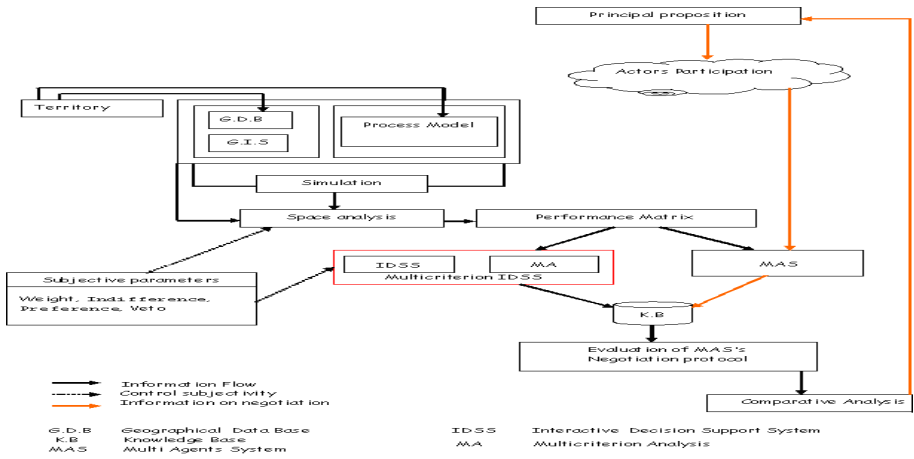


Fig. 1. The suggested decisional model

*The Multicriterion Model.* In order to choose the most adequate solution, the multicriterion classification of the various actions is, then, made by the use of a sorting approach (ordinal or nominal). This analysis performs a partial aggregation well suitable to the land management. Indeed, it can treat criteria extremely diverse (economic, social, ecological,...) and of qualitative and quantitative type [6]. Also, it considers many subjective variables (weight, veto, indifference, preference,...) making possible to the decision maker expressing his preferences.

*The Negotiation Model.* The decision-making model of Simon [11] as well as other extensions neglects three key elements of the decision-making in a spatial context related to the presence of multiple actors: the participation, the negotiation and the consultation. Several ways can be borrowed to integrate the antagonistic points of view of several actors. The "Multi-Agent Systems" MAS prove particularly adapted to the modeling and the simulation of this situation [2]. Modeling by the MAS aims at representing the multiplicity of the actors, their diversity, their behavior like their interaction.

## 7 Experimental Study

The objective of this setting in situation is to support the methodological proposals by a confrontation with the tools and the data at disposal. The problem of localization clearly defined and approached by Joerin [8] using the Tricotomic Segmentation (independent criteria) constitutes an ideal context to test the application of our model. In order to evaluate the performance of our model, we illustrate our step on the same example proposed by [8] and which consists at establishing a land suitability map for habitation (Where construct?). The choice of the area test results from the great number of space data at disposal. Our

model is dynamically used according to the procedure proposed by Pictet [10]. It includes three principal phases:

*The Structuring of the Model.* In this phase, we consider the different means at disposal in the area of study: geographic chart, data and evaluation methods.

1. Delimitation of the area of study: situated in the canton of Vaud, to approximately 15 km in the north of Lausanne. The surface of this area is of 52.500 km<sup>2</sup>.
2. Identification of the actions: a total of 650 zones (actions) cover entirely the studied area. Limiting reference actions<sup>3</sup> are also defined.
3. Identification and evaluation of the criteria: the delimitation of the area of study influences the choice of the criteria as well as their evaluation methods.

**Table 1.** The criteria evaluation table

Criteria	Type	Factors(sub-Criteria)	Evaluation Method
<b>1</b> Harm	Natural	Air Pollutions, Odors	Attribution of a note
<b>2</b> Noise	Social	Motorways, Railways	Attribution of a note
<b>3</b> Impacts	Social	Sectorial plan	Attribution of a note
<b>4</b> Geotechnical and Natural Risks	Natural	Constraints,Landslides, Flood,Seism,Firescriptsiz	Procedures of space analysis Consultation of the experts
<b>5</b> Equipment	Economic	Distance to: Gas, Electricity,Water, Roads	Balanced distances for the various networks
<b>6</b> Accessibility	Social	Distcance to localities	Attribution of a note
<b>7</b> Climate	Natural	Sun, Fog, Temperature	Attribution of a note

*The Exploitation of the Model.* It is the analytical part of the process. The spatial analysis allows evaluating the criteria and the multicriterion sorting analysis (the preferences aggregation) classifies the actions into categories. To treat the considered problem, we apply an ordinal sorting (the procedure is presented in section 5.2) to the set of the identified actions. The actions are classified in three categories of suitability. The low category  $A_1$  constituted by actions issued too bad, the category  $A_3$  gathering actions issued sufficiently good (actions which define the required site) and the category  $A_2$  containing the actions which can be classified neither in  $A_1$ , nor in  $A_3$ . The actions belonging to  $A_2$  facilitate the task of the required site limit determination. To treat the second problem, the decision maker can choose the various types of land use, then defines for each type a set of prototypes. It is enough later to apply a nominal sorting (the procedure is presented in section 5.3) assigning each action to a type of land use.

*The Concretization of the Results.* It aims primarily at the social acceptance of the result.

---

<sup>3</sup> The actions of reference are used as the limits to the categories to which the potential actions will be affected.

*Experimental Results, Discussion.* In this case study, the criterion 5 (*Equipment*) and the criterion 6 (*Accessibility*) are dependent. They are positively correlated, the calculated coefficient of correlation is  $r_{5,6} = +0.60047$ . The presence of the one implies the other and to suppose that these criteria are independent can harm the final decision because of the data redundancy . The obtained results are checked against those obtained in [8] considering the same reference actions and the same subjective parameters. The classification resulted from a multicriterion aggregation considering that the criteria (*Equipment* and *Accessibility*) are independent by the use of a simple arithmetic average. The concessions granted by the study made by [8] accept that the edge of the motorway is qualified as doubtful. However, our multicriterion aggregation model using the Choquet’s integral dealt with the dependence between these two criteria and decided that the doubtful zones near the motorway are bad. Consequently, the numbers of the good and doubtful actions dropped considerably (Figure 2).

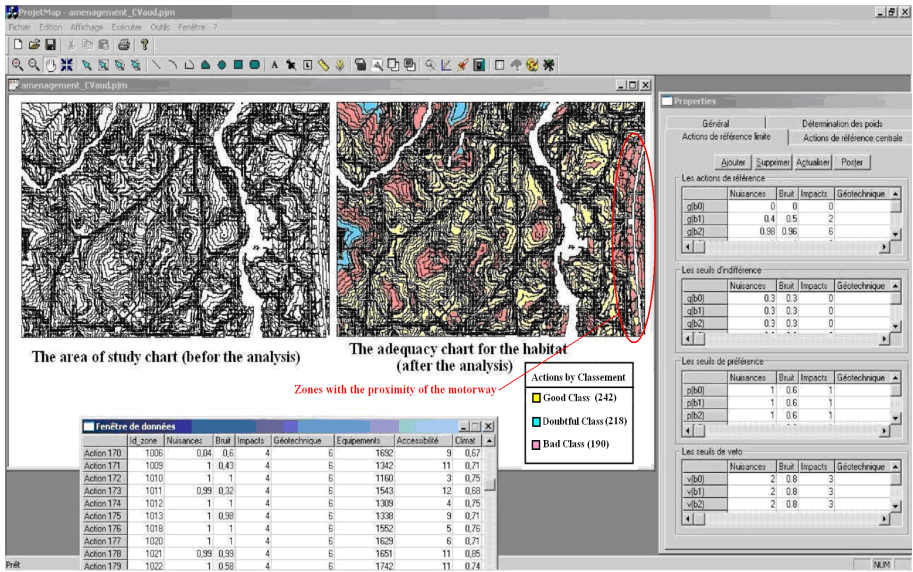


Fig. 2. Displaying of the multicriterion analysis results using the suggested model

## 8 Conclusion

In this paper, we have addressed the problem of interactive criteria characterizing human subjectivity in the multicriterion decision making. We have proposed an efficient constructive approach based on a fuzzy measurement model able to take into account the interactions between criteria. Our contribution agrees with the idea that using Choquet integral operator provides an experimental setting that permits to optimize the quality of decision in Territory Planning. Moreover, the ultimate goals of this interdisciplinary study were to: Yield a



precise description of the environmental project; evaluate and compare multiple scenarios to determine the best solutions and allow the participation of multiple parties with conflicting view points. The obtained results are of a theoretical, methodological and algorithmic order. However, the principal limitation of the evolution of this type of procedures of information treatment resides in the policy of the data diffusion and much of potential users do not launch out in this type of step.

## References

1. Choquet, G.: Theory of capacities. *Annales de l'institut Fourier* 5, 131–295 (1953)
2. Chuan Jiang, Y., Jiang, J.C.: A multi-agent coordination model for the variation of underlying network topology. *Expert systems with applications* 29, 372–382 (2005)
3. Davignon, G., Sauvageau, M.: L'aide Multicritère à la décision: un cas d'intégration des critères techniques et environnementaux à Hydro-Quebec. CER-ADO, CANADA (1994)
4. Eastman, J.R., Toledaro, J.: Exploration in Geographic Information Systems Technology. *GIS and Decision Making* 4 (1994)
5. Grabisch, M.: k-order additivity discrete fuzzy measure and their representation. *Pattern Recognition Letters* 92, 167–189 (1997)
6. Hamdadou, D., Labed, K.: Un Processus Décisionnel Par Utilisation Des SIG Et Des Méthodes Multicritères Pour l'Aménagement Du Territoire: PRODUSMAT. In: MCSEAI 2006, pp. 671–676 (2006)
7. Hamdadou, D., Labed, K.: Proposal for a Decision-making process in Regional planning: GIS, Multicriterion Approach, Choquet's Integral and Genetic Algorithms. In: ERIMA 2007 (to appear, 2007)
8. Joerin, F.: Décider sur le territoire: Proposition d'une approche par l'utilisation de SIG et de MMC. Thèse de Doctorat, Ecole Polytechnique Fédérale de Lausanne (1997)
9. Marichal, J.L.: Determination of weights of interacting criteria from a reference. *European journal of operational Research* 24, 641–650 (2003)
10. Pictet, J.: Dépasser l'évaluation environnementale, procédure d'étude et insertion dans la décision globale. Collection Meta, Presses Polytechniques et universitaires Romandes 1015 (1996)
11. Roy, B., Bouyssou, D.: Aide multicritère à la décision: Méthodes et cas. *Economica* (1993)
12. Scharling, A.: Pratiquer ELECTRE et Prométhée: un complément à décider sur plusieurs critères. Presses polytechniques et universitaires Romandes 1015 (1996)
13. Sugeno, M.: Theory of fuzzy integrals and its applications. Thèse de Doctorat, Institut de technologie de Tokyo, Japan (1974)

# Radial Basis Function Neural Network Based on Order Statistics

Jose A. Moreno-Escobar<sup>1</sup>, Francisco J. Gallegos-Funes<sup>1</sup>, Volodymyr Ponomaryov<sup>2</sup>,  
and Jose M. de-la-Rosa-Vazquez<sup>1</sup>

National Polytechnic Institute of Mexico  
Mechanical and Electrical Engineering Higher School

<sup>1</sup> Av. IPN s/n, U.P.A.L.M. SEPI-ESIME, Edif. Z, Acceso 3, Tercer Piso,  
Col. Lindavista, 07738, Mexico, D. F., Mexico  
fgallegos@ipn.mx

<sup>2</sup> Av. Santa Ana 1000, Col. San Francisco Culhuacan, 04430, Mexico, D. F., Mexico  
vponomar@ipn.mx

**Abstract.** In this paper we present a new type of Radial Basis Function (RBF) Neural Network based in order statistics for image classification applications. The proposed neural network uses the Median M-type (MM) estimator in the scheme of radial basis function to train the neural network. The proposed network is less biased by the presence of outliers in the training set and was proved an accurate estimation of the implied probabilities. From simulation results we show that the proposed neural network has better classification capabilities in comparison with other RBF based algorithms.

## 1 Introduction

The neural networks represent a technology that is rooted in many disciplines: mathematics, statistics, physics, computer science, neuroscience, engineering, and medicine [1]. They are particularly useful in problems where decision rules are vague and there is no explicit knowledge about the probability density functions governing sample distributions [1-3]. The neural networks find applications in modeling, time series analysis, pattern recognition, and signal processing [1].

The Radial Basis Function (RBF) network involves three layers with entirely different roles [4-7]. The input layer is made up of source nodes that connect the network to its environment. The second layer is the only hidden layer in the network, applies a nonlinear transformation from the input space to the hidden space. The output layer is linear, supplying the response of the network to the activation signal or pattern applied to the input layer

In this paper we propose a new type of RBF neural network based on order statistics for image classification purposes. The neural network uses the Median M-Type (MM) estimator [8] in the scheme of radial basis function to train the neural network according with the schemes found in the references [9,10].

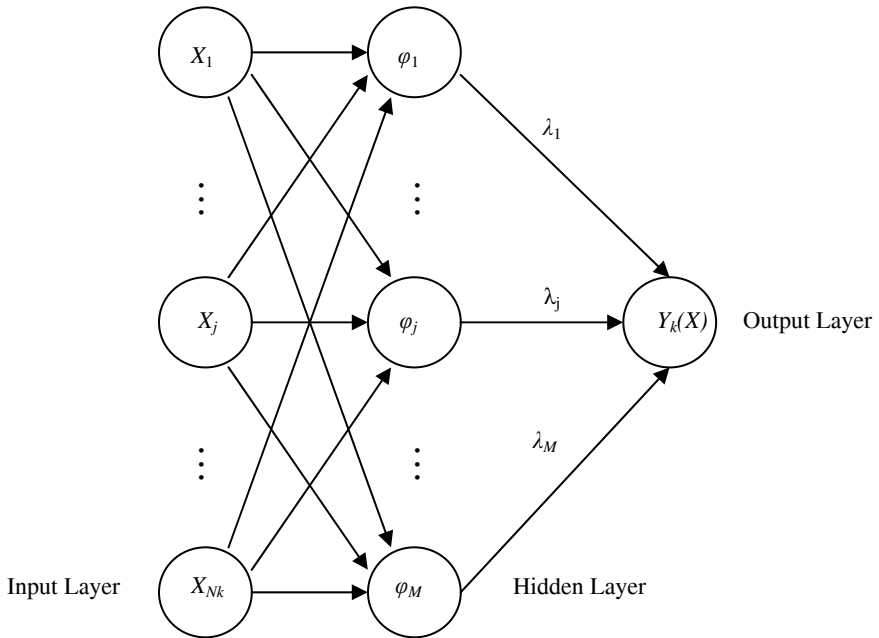
The rest of this paper is organized as follows. An overview of The RBF neural network is presented in section 2. In section 3 we formulate the proposed MMRBF

neural network. Experimental results in the artificial data classification and the mam-mographic image analysis for the proposed algorithm and other RBF based networks are presented in section 4. Finally, we draw our conclusions in section 5.

## 2 Overview of Radial Basis Function Neural Network

The RBF have their fundamentals drawn from probability function estimation theory [4]. The Radial Basis Functions (RBF) have been used in several applications for pattern classification and functional modeling [5]. These functions have been found to have very good functional approximation capabilities [5].

In the RBF neural networks each network input is assigned to a vector entry and the outputs correspond either to a set of functions to be modeled by the network or to several associated classes [1,6,7]. The structure of the RBF neural network is presented in Figure 1. From this Figure, each of  $N_k$  components of the input vector  $\mathbf{X}$  feeds forward to  $M$  basis functions whose outputs are linearly combined with weights  $\{\lambda_j\}_{j=1}^M$  into the network output  $Y_k(\mathbf{X})$



**Fig. 1.** Structure of Radial Basis Function Neural Network

Several functions have been tested as activation functions for RBF neural networks. In pattern classification applications the Gaussian function is preferred [9,10],

$$\phi_j(\mathbf{X}) = \exp\left[-(\mu_j - \mathbf{X})^T \Sigma_j^{-1} (\mu_j - \mathbf{X})\right] \quad (1)$$

where  $\mathbf{X}$  is the input feature vector,  $\mu_j$  is the mean vector and  $\Sigma_j$  is the covariance matrix of the  $j$ th Gaussian function. Geometrically,  $\mu_j$  represents the center or location and  $\Sigma_j$  the shape of the basis functions. Statistically, an activation function models a probability density function where  $\mu_j$  and  $\Sigma_j$  represent the first and second order statistics. A hidden unit function can be represented as a hyper-ellipsoid in the  $N$ -dimensional space.

The output layer implements a weighted sum of hidden-unit outputs [9,10]:

$$\psi_k(\mathbf{X}) = \sum_{j=1}^L \lambda_{jk} \phi_j(\mathbf{X}) \quad (2)$$

where  $L$  is the number of hidden units,  $M$  is the number of outputs with  $k=1, \dots, M$ . The weights  $\lambda_{kj}$  show the distribution of the hidden unit  $j$  for modeling the output  $k$ .

Radial Basis Functions have interesting properties which make them attractive in several applications. A combined unsupervised-supervised learning technique has been used in order to estimate the RBF parameters [6]. In the unsupervised stage,  $k$ -means clustering algorithm is used to find the pdf's parameters, LMS or instead pseudo-inverse matrix can be used in the supervised stage to calculate the weights coefficients in the neural network [6,9,10].

### 3 Median M-Type Radial Basis Function Neural Network

We present the use of the MM-estimator with different influence functions as statistic estimation in the Radial Basis Function network architecture. The proposed network is called as Median M-type Radial Basis Function (MMRBF) neural network.

The activation function used in the proposed network is the inverse multiquadratic function [1,2,6]:

$$\phi_j(\mathbf{X}) = \frac{1}{\sqrt{\mathbf{X}^2 + \beta_j^2}} \quad (3)$$

where  $\mathbf{X}$  is the input feature vector,  $\beta_j$  is a real constant. In our simulation results  $\beta_j=1$ .

In our case we use the clustering  $k$ -means algorithm to estimate the parameters of the MMRBF neural network [1,2]. The  $k$ -means algorithm is used in the unsupervised stage. The input feature vector  $\mathbf{X}$  is classified in  $k$  different clusters. A new vector  $\mathbf{x}$  is assigned to the cluster  $k$  whose centroid  $\mu_k$  is the closest one to the vector. The centroid vector is updated according to,

$$\mu_k = \mu_k + \frac{1}{N_k}(\mathbf{x} - \mu_k) \quad (4)$$

where  $N_k$  is the number of vectors already assigned to the  $k$ -cluster. The centroids can be updated at the end of several iterations or after the test of each new vector. The centroids can be calculated with or without the new vector. By other hand, the steps for the  $k$ -means algorithm are the following:

Step 1	Select an initial partition with $k$ clusters. Repeat steps 2 through 4 until the cluster membership stabilizes.
Step 2	Generate a new partition by assigning each pattern to its closest cluster center.
Step 3	Compute new cluster centers as the centroids of the clusters.
Step 4	Repeat steps 2 and 3 until an optimum value of the criterion function is found.

The Median M-type (MM) estimator is used in the proposal RBF neural network [8]. The non-iterative MM-estimator used as robust statistics estimate of a cluster center is given by,

$$\mu_k = \text{med}\{\mathbf{X}\varphi(\mathbf{X} - \theta)\} \quad (5)$$

where  $\mathbf{X}$  is the input data sample,  $\varphi$  is the normalized influence function  $\psi : \psi(\mathbf{X}) = \mathbf{X}\varphi(\mathbf{X})$ ,  $\theta = \text{med}\{X_k\}$  is the initial estimate, and  $k=1, 2, \dots, N_k$ .

In our experiments we use the following influence functions [8]:

the simple cut (skipped mean) influence function,

$$\psi_{\text{cut}(r)}(X) = X \cdot 1_{[-r,r]}(X) = \begin{cases} X, & |X| \leq r \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

and the Tukey biweight influence function,

$$\psi_{\text{bi}(r)}(X) = \begin{cases} X^2(r^2 - X^2), & |X| \leq r \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

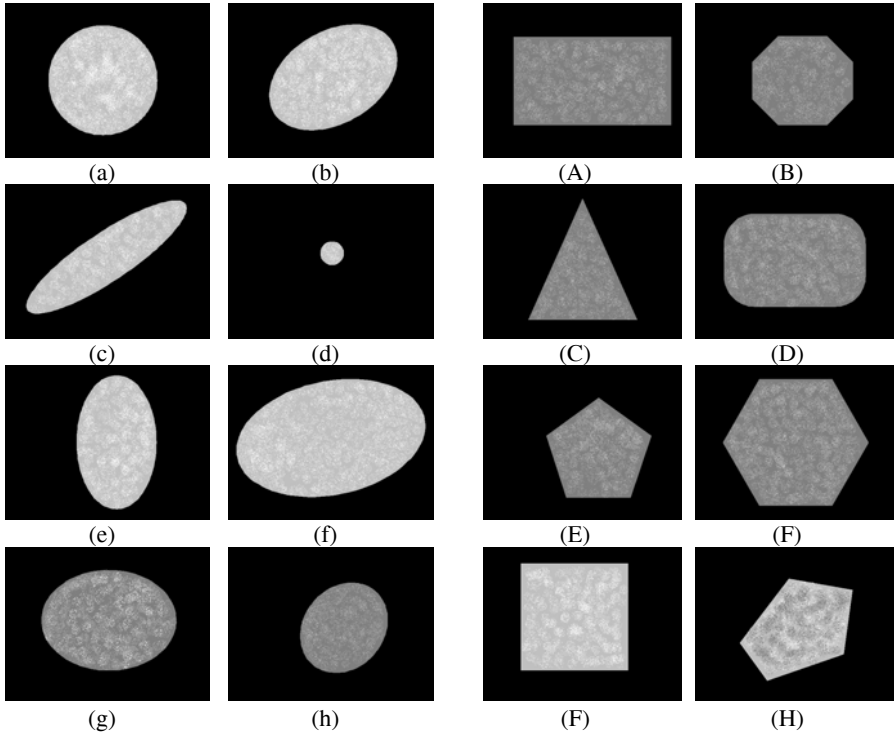
where  $X$  is a data sample and  $r$  is a real constant. The parameter  $r$  depends of the data to process and can be change for different influence functions.

## 4 Experimental Results

The described MMRBF neural network has been evaluated, and his performance has been compared with the Simple RBF,  $\alpha$ -Trimmed Mean RBF, and Median RBF neural networks [5,9,10].

Experiment 1: Artificial data classification.

Figure 2 shows the images used to train the proposed MMRBF neural network and other networks used as comparative. In this figure, the first six images of Group A have common texture or filling which are different form the six first images of Group B. The last two images of each group have a texture or filling that is similar to the opposite group, that is, the last two images of Group A have similar filling than the images of Group B, and vice versa. The main idea here of using textures in figures is to try to simulate medical image textures. To train the networks for getting the appropriate pdf's parameters were used 10 images (five of each group) as the ones shown in Figure 2. The objective of the experiment is to classify between 2 main groups.



**Fig. 2.** Proof images used to train the neural networks, Group A contains circles and ellipses denoted with small letters, Group B contains many kinds of polygons denoted with capital letters

In the segmentation stage was obtained three numerical data or features, which are the compactness, average gray value, and standard deviation [11-13].

Having the images, the first step is extracting numerical data from them. Afterwards we determined the center of the activation functions. The number of elements used in each activation function depends on the algorithm implemented. The number of elements used to train the comparative Simple RBF,  $\alpha$ -Trimmed Mean RBF, and Median RBF neural networks varies in accordance to the training algorithms found in references [5,9,10]. In the case of the proposed MMRBF we use eq. (5) in combination with eq. (6) and (7) to determine the elements to be used.

The training results for different networks are shown in Table 1. In the probe stage we use 30 images (15 of each group), these images are of different form that the images used in the training stage. Figure 3 presents some images used in the probe stage. The results obtained are shown in Table 2.

From Tables 1 and 2 we can appreciate that the difference between algorithms is not big, and that percentages of efficiency, uncertainty and error vary from training stage to probe stage.

**Table 1.** Experimental results obtained with different RBF algorithms in training stage

		Group A	Group B	Total
SIMPLE RBF	Efficiency	60%	100%	80%
	Uncertainty	0%	0%	0%
	Error	40%	0%	20%
MEDIAN RBF	Efficiency	60%	100%	80%
	Uncertainty	0%	0%	0%
	Error	40%	0%	20%
$\alpha$ -TRIMMED MEAN RBF	Efficiency	80%	100%	90%
	Uncertainty	0%	0%	0%
	Error	20%	0%	15%
MMRBF Simple Cut	Efficiency	67%	100%	83.5%
	Uncertainty	0%	0%	0%
	Error	33%	0%	16.5%
MMRBF Tukey	Efficiency	67%	100%	83.5%
	Uncertainty	0%	0%	0%
	Error	33%	0%	16.5%

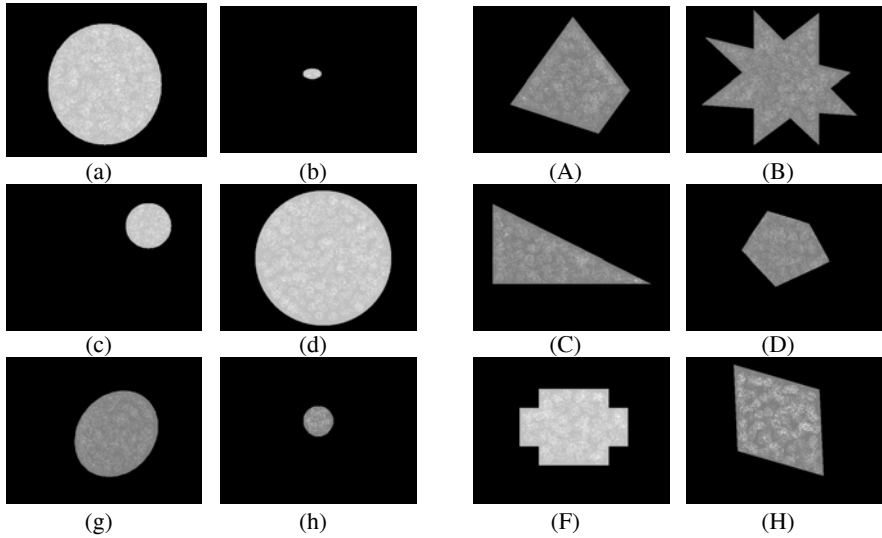
**Fig. 3.** Some images used in probe stage

Table 3 shows the comparison between the RBF algorithms implemented here. We can see from this Table that  $\alpha$ -TRIMMED MEAN RBF has the best efficiency in training stage, but in probe stage the best results are given by the proposed MMRBF neural network.

**Table 2.** Results obtained with different RBF algorithms in probe stage

		Group A	Group B	Total
SIMPLE RBF	Efficiency	67%	93%	80%
	Uncertainty	0%	0%	0%
	Error	33%	7%	20%
MEDIAN RBF	Efficiency	67%	93%	80%
	Uncertainty	0%	0%	0%
	Error	33%	7%	20%
$\alpha$ -TRIMMED MEAN RBF	Efficiency	67%	87%	77%
	Uncertainty	0%	0%	0%
	Error	33%	13%	23%
MMRBF Simple Cut	Efficiency	67%	100%	83.5%
	Uncertainty	0%	0%	0%
	Error	33%	0%	16.5%
MMRBF Tukey	Efficiency	67%	100%	83.5%
	Uncertainty	0%	0%	0%
	Error	33%	0%	16.5%

**Table 3.** Experimental results of efficiency between different RBF algorithms

Neural Networks	SIMPLE RBF	MEDIAN RBF	$\alpha$ -TRIMMED MEAN RBF
Training stage			
MMRBF Simple Cut	3.5%	3.5%	-6.5%
MMRBF Tukey	3.5%	3.5%	-6.5%
Probe stage			
MMRBF Simple Cut	3.5%	3.5%	6.5%
MMRBF Tukey	3.5%	3.5%	6.5%

Experiment 2: Mammographic image analysis.

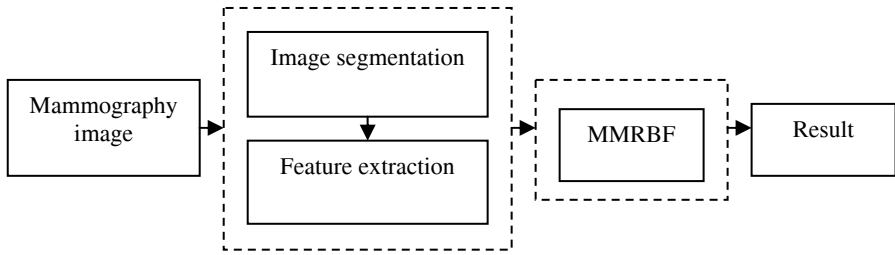
The images used to train the proposed MMRBF neural network and other networks used as comparative were obtained from Mammographic Image Analysis Society (MIAS) web site [14]. This image collection is constituted by 322 images divided in three categories: normal, benign, and malign.

To train the networks for getting the appropriate pdf's parameters was used 32 images: 8 normal, 8 with benign abnormalities, 8 with malign abnormalities, 4 with benign microcalcifications and 4 with malign microcalcifications [15,16]. The objective of the experiment is to classify between 2 main groups: Group A contains normal and benign abnormalities and Group B contains malign abnormalities and any kind of microcalcification.

The classification process is shown in Figure 4. Having the mammography image, we proceed to segment it in 2 main regions of interest [15,16]:

- Region 1 is constituted of a detected object, and
- Region 2 is constituted of the breast, without the detected object.





**Fig. 4.** Block diagram of proposed MMRBF neural network

The segmentation results are shown in Figure 5. After segmentation, the following consists in extracting numerical data from the regions of interest. Here we obtained eight numerical data or features, which are compactness, contrast, standard deviation of detected object and breast, average value of detected object and breast, and range of values of detected object and breast [11-13].



**Fig. 5.** Image segmentation results

To probe the performance of the networks was used 125 images: 40 normal (NORMAL), 38 with benign abnormalities (AN\_BEN), 30 malign abnormalities (AN\_MAL), 8 with benign microcalcifications (uC\_BEN), and 9 with malign microcalcifications (uC\_MAL).

In Table 4 we present the performance results in terms of efficiency, uncertainty, and error for different neural networks. From this Table one can see that the proposed MMRBF neural network provides the best results in comparison with other RBF based networks in the most of cases.

Table 5 presents the performance results in terms of efficiency in each group of images (Group A and Group 2). In this Table one can see that the best results are obtained when we use the proposed MMRBF neural network.

Table 6 show the comparison between different RBF algorithms used in the mammographic image analysis. We observe from this Table that the proposed MMRBF neural network has the best efficiency in the probe stage in the most of the cases.

**Table 4.** Performance results in the MIAS collection by use different neural networks

Neural Networks		NORMAL	AN_BEN	AN_MAL	uC_BEN	uC_MAL	TOTAL
SIMPLE RBF	Efficiency	52.50%	47.37%	30.00%	12.50%	55.56%	39.59%
	Uncertainty	1.33%	0.00%	0.00%	0.00%	0.00%	0.27%
	Error	46.17%	52.63%	70.00%	87.50%	44.44%	60.15%
MEDIAN RBF	Efficiency	65.00%	60.53%	26.67%	12.50%	66.67%	46.27%
	Uncertainty	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	Error	35.00%	39.47%	73.33%	87.50%	33.33%	53.73%
$\alpha$ -TRIMMED MEAN RBF	Efficiency	47.50%	57.89%	50.00%	87.50%	55.56%	59.69%
	Uncertainty	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	Error	52.50%	42.11%	50.00%	12.50%	44.44%	40.31%
SIMPLE CUT MMRBF	Efficiency	70.00%	71.05%	30.00%	62.50%	88.89%	64.49%
	Uncertainty	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	Error	30.00%	28.95%	70.00%	37.50%	11.11%	35.51%
TUKEY MMRBF	Efficiency	70.00%	57.89%	36.67%	37.50%	77.78%	55.97%
	Uncertainty	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	Error	30.00%	42.11%	63.33%	62.50%	22.22%	44.03%

**Table 5.** Efficiency results by group of different Neural Networks

Neural Networks	Group 1			Group 2		
	Efficiency	Uncertainty	Error	Efficiency	Uncertainty	Error
SIMPLE RBF	49.93%	0.67%	49.40%	32.69%	0.00%	67.31%
MEDIAN RBF	62.77%	0.00%	37.23%	35.28%	0.00%	64.72%
$\alpha$ -TRIMMED MEAN RBF	52.70%	0.00%	47.30%	64.35%	0.00%	35.65%
SIMPLE CUT MMRBF	70.53%	0.00%	29.47%	60.46%	0.00%	39.54%
TUKEY MMRBF	63.95%	0.00%	36.05%	50.65%	0.00%	49.35%

**Table 6.** Efficiency results between the MMFBR and the algorithms used as comparative

Neural Networks	SIMPLE RBF	MEDIANA RBF	$\alpha$ -TRIMMED MEAN RBF
SIMPLE CUT MMRBF	24.90%	18.22%	4.80%
TUKEY MMRBF	16.38%	9.70%	-3.72%

To evaluate the performance of the neural networks in terms of medical purposes, there were calculated two quantities: sensitivity and specificity [17].

Sensitivity is the probability that a medical test delivers a positive result when a group of patients with certain illness is under study [17],

$$S_n = TP / (TP + FN) \tag{8}$$

where  $S_n$  is sensitivity,  $TP$  is the number of true positive that are correct and  $FN$  is the number of false negatives, that is, the negative results that are not correct.

Specificity is the probability that a medical test delivers a negative result when a group of patients under study do not have certain illness [17],

$$Sp = TN / (TN + FP) \quad (9)$$

where  $S_p$  is specificity,  $TN$  is the number of negative results that are correct and  $FP$  is the number of false positives, that is, the positive results that are not correct.

Table 7 shows the sensitivity and specificity values obtained for every one of the Neural Networks. It can be appreciated that the specificity of the proposed MMRBF using simple cut influence function is the highest one, about a 20% above  $\alpha$ -TRIMMED MEAN RBF, but this last network has the best sensitivity, about 10% above the mentioned MMRBF Simple Cut.

**Table 7.** Sensitivity and specificity results for different Neural Networks

Neural Networks	Sensitivity	Specificity
SIMPLE RBF	34.04%	50.00%
MEDIAN RBF	31.91%	62.82%
$\alpha$ -TRIMMED MEAN RBF	57.45%	52.56%
SIMPLE CUT MMRBF	48.93%	70.51%
TUKEY MMRBF	44.68%	64.10%

## 5 Conclusions

We present the novel MMRBF neural network; it uses the MM-estimator in the scheme of radial basis function to train the proposed neural network. The experimental results in the case of artificial data classification and mammographic image analysis obtained with the use of the proposed MMRBF are better than the results obtained with other RBF based algorithms.

Unfortunately the error is still big in the case of mammographic image analysis, it is due to simple segmentation algorithm.

As future work we will probe with other segmentation algorithms to improve the classification of the mammographic images.

## Acknowledgements

The authors thank the National Polytechnic Institute of Mexico for its support.

## References

1. Haykin, S.: Neural Networks, a Comprehensive Foundation. Prentice Hall, NJ (1994)
2. Rojas, R.: Neural Networks: A Systematic Introduction. Springer, Berlin (1996)

3. Egmont-Petersen, M., de Ridder, D., Handels, H.: Image processing with neural networks - a review. *Pattern Recognition* 35, 2279–2301 (2002)
4. Buhmann, M.D.: *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics (2003)
5. Musavi, M.T., Ahmed, W., Chan, K.H., Faris, K.B., Hummels, D.M.: On the training of radial basis function classifiers. *Neural Networks* 5, 595–603 (1992)
6. Karayiannis, N.B., Weiqun Mi, G.: Growing radial basis neural networks: merging supervised and unsupervised learning with network growth techniques. *IEEE Trans. Neural Networks* 8(6), 1492–1506 (1997)
7. Karayiannis, N.B., Randolph-Gips, M.M.: On the construction and training of reformulated radial basis function neural networks. *IEEE Trans. Neural Networks*. 14(4), 835–846 (2003)
8. Gallegos, F., Ponomaryov, V.: Real-time image filtering scheme based on robust estimators in presence of impulsive noise. *Real Time Imaging*. 8(2), 78–90 (2004)
9. Bors, A.G., Pitas, I.: Median radial basis function neural network. *IEEE Trans. Neural Networks* 7(6), 1351–1364 (1996)
10. Bors, A.G., Pitas, I.: Object classification in 3-D images using alpha-trimmed mean radial basis function network. *IEEE Trans. Image Process* 8(12), 1744–1756 (1999)
11. González, R.C., Woods, R.E.: *Tratamiento Digital de Imágenes*. Addison Wesley, Díaz de Santos (1996)
12. Ritter, G.: *Handbook of Computer Vision Algorithms in Image Algebra*. CRC Press, Boca Raton-New York (2001)
13. Myler, H.R., Weeks, A.R.: *The Pocket Handbook of Image Processing Algorithms in C*. Prentice-Hall, Englewood Cliffs (1993)
14. <http://www.wiau.man.ac.uk/services/MIAS/MIAScom.html>
15. Webb, G.: *Introduction to Biomedical Imaging*. Wiley-IEEE Press, New Jersey (2002)
16. Suri, J.S., Rangayyan, R.M.: *Recent Advances in Breast Imaging, Mammography, and Computer-Aided Diagnosis of Breast Cancer*. SPIE Press, Bellingham (2006)
17. <http://www.cmh.edu/stats/definitions/>

# Temperature Cycling on Simulated Annealing for Neural Network Learning

Sergio Ledesma, Miguel Torres, Donato Hernández, Gabriel Aviña,  
and Guadalupe García

Dept. of Electrical & Computer Engineering, University of Guanajuato.  
Salamanca, Gto. 36700, México

{selo, mtorres, donato, avina, garciag}@salamanca.ugto.mx

**Abstract.** Artificial neural networks are used to solve problems that are difficult for humans and computers. Unfortunately, artificial neural network training is time consuming and, because it is a random process, several cold starts are recommended. Neural network training is typically a two step process. First, the network's weights are initialized using a no greedy method to elude local minima. Second, an optimization method (i.e., conjugate gradient learning) is used to quickly find the nearest local minimum. In general, training must be performed to reduce the mean square error computed between the desired output and the actual network output. One common method for network initialization is simulated annealing; it is used to assign good starting values to the network's weights before performing the optimization. The performance of simulated annealing depends strongly on the cooling process. A cooling schedule based on temperature cycling is proposed to improve artificial neural network training. It is shown that temperature cycling reduces training time while decreasing the mean square error on auto-associative neural networks. Three auto-associative problems: The Trifolium, The Cardioid, and The Lemniscate of Bernoulli, are solved using exponential cooling, linear cooling and temperature cycling to verify our results.

## 1 Introduction

Artificial neural networks (ANNs) can be used to solve complex problems where noise immunity is important. There are two ways to train an ANN: supervised and un-supervised training. Supervised training requires a *training set* where the input and the desired output of the network are provided for several training cases. On the other hand, un-supervised training requires only the input of the network, and the ANN is supposed to classify (separate) the data appropriately.

When the desired output of the network is equal to the input of the network for each training case, the ANN is known as an auto-associative neural network. Alternatively, when the ANN is used to separate objects based on their properties, the ANN is known as a classifier. Lastly, when the ANN is used to map a specific input to a desired output, the network is known as a generic or a mapping neural network. This paper is focused on auto-associate neural networks

under supervised learning. To verify our results, the classical curves of Figure 1 were used for training three different auto-associate neural networks. The mean squared error (mse) and the training time were measured to compare our method with a common ANN initialization method.

In order for an ANN to learn, a *training set* is required, that is, for each curve in Figure 1 an appropriate *training set* containing the curve at all possible rotation angles needs to be built. The design of the training is very important because an undesired ANN training effect known as overfitting may result, see [10]. Basically, if the number of training cases is not big enough compare with the number of network's weights, the training process may not have enough cases to adjust appropriately the network's weights, that is, the network's weights may not be good for other data but the *training set*. Thus, an ANN with more neurons (more weights) requires training sets with more training cases, and more training cases means a longer training time. Because of this, training methods that can train ANNs fast and with little error are required. We propose a method that can reduce the training time on auto-associate neural networks.

This paper is organized as follows. In Section 2, background information about ANN training is reviewed. In Section 3, a new cooling schedule for simulated annealing is presented. In Section 4, simulation experiments are performed to verify our results. Finally, Section 5 presents some conclusions and direction for future work.

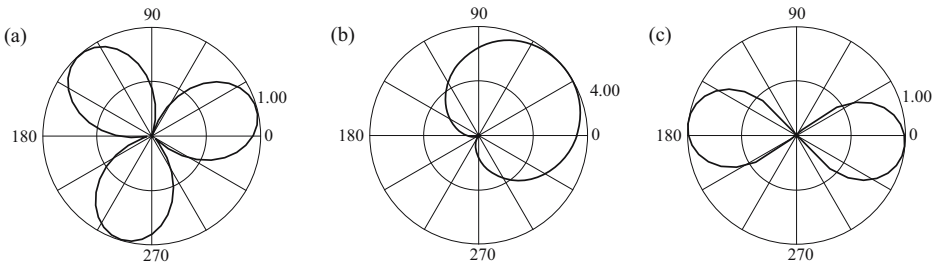


Fig. 1. (a) The Trifolium, (b) The Cardioid, and (c) The Lemniscate of Bernoulli

## 2 Background Information

### 2.1 Neural Network Learning

Neural networks are useful when modeling the data is difficult or impossible. For these cases, neural networks can be used to learn from a data set known as the training set; and these networks usually may not have any specific information about the data source.

During a process call training, the network's weights are adjusted until the actual network's output and a desired output are as close as possible, see [13]. During this process, it is valid to re-design the network structure adjusting the number of layers or neurons until a specified performance is obtained. Once the

network has been trained, it must be evaluated using a different training set, called the *validation set*. Several heuristic approaches might be used until the *validation set* and the *training set* perform similarly, see [7].

The training process can be divided in two steps: initialization and optimization. The initialization process might be something as simple as assigning initial random values to the network's weights [12] or something much more sophisticated such as: genetic algorithms, simulated annealing, or regression. The optimization process is usually a gradient based algorithm, and it requires a good starting point to be successful. This means that the complete training process depends on both, the initialization process and the optimization process.

## 2.2 The Method of Simulated Annealing (SA)

At the heart of the method of simulated annealing is an analogy with thermodynamics, specifically with the way that liquids freeze and crystallize, or metals cool and anneal [11]. The method of simulated is an optimization method that tries to imitate the natural annealing process which occurs when a material is heated and then cooled down in a controlled manner. One classic problem solved by SA is the Traveling Salesman Problem, see [4]. Contrary to other optimization methods, SA is a no greedy optimization method and, hence it does not fall easily into local minima. One of the most important practical considerations when implementing the method of SA is to use a high quality random generator, see [6] and [11]. For ANN training, the method of SA perturbs randomly the network's weights following a cooling schedule. Once the network's weights have been perturbed, the performance of the neural network is evaluated using an appropriate *training set*. In general, the cooling schedule may be linear or exponential, and may iterate at each temperature or increase the number of iterations at a specific temperature when improvement occurs, see [1], [2] and [6].

Each time a new solution has been perturbed and evaluated, the oracle makes a decision about whether the new solution is accepted or rejected using the metropolis algorithm [3] and [11]. Some implementations of SA accept a new solution only if the new solution is better than the old one, i.e. it accepts the solution only when the mse decreases; see [6]. However, it is always more efficient to accept the solution even when the new solution has not less error than the previous solution. The probability to accept a new solution was first incorporated into numerical calculations by Metropolis [8] as shown

$$P_a(\Delta E, y) = \begin{cases} e^{-\frac{k\Delta E}{y}}, & \Delta E > 0 \\ 1, & \Delta E \leq 0, \end{cases} \quad (1)$$

where

$$\begin{aligned} \Delta E &= Error_{new\ solution} - Error_{current\ solution} \\ y &= \text{Current temperature,} \end{aligned}$$

and  $k$  is the acceptance constant that depends on the range of the network's weights and the network's input. Thus, at high temperatures, the algorithm

may frequently accept a solution even if it is not better than the previous one [3]. During this phase, the algorithm explores in a very wide range looking for an optimal solution, and it is not concerned with the quality of the solution. As the temperature decreases, the algorithm is more selective, and it accepts a new solution only if its error is less than or very similar to the previous solution error following the decision made by the oracle.

In the next section, a cooling schedule will be proposed for implementing SA for ANN's initialization.

### 3 Proposed Method

One key factor when training multi-layer feed forward neural networks using SA for initialization is to choose an appropriate acceptance constant,  $k$  in Equation [1]. This acceptance constant depends directly on the temperature range, the training set, and the network's weight allowed values. Failing to take into consideration these factors may affect adversely ANN's learning when using SA or other training methods.

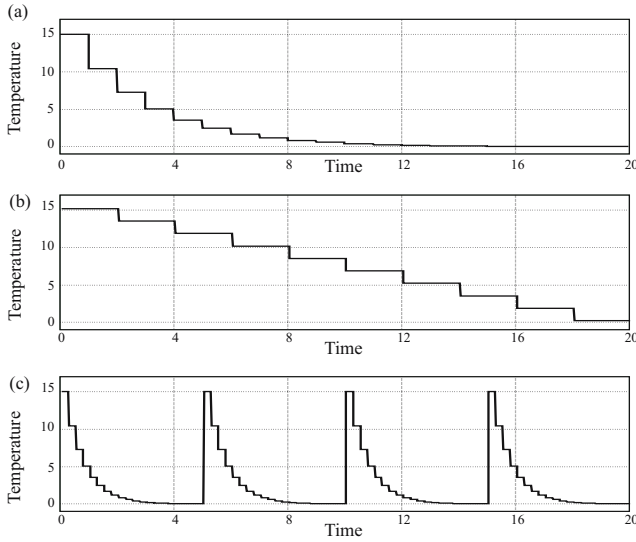
During the preparation of the *training set* is important to scale the data appropriately. In general, it is convenient to restrict the network's input data in the range from  $-3$  to  $3$  (or less), resulting in network's weights in the range from  $-10$  to  $10$  or so. Increasing the network's input range more is not recommended for SA, because a wider input range means a wider weights' range and, therefore more combinations for SA, which results on long trainings. To simplify the implementation of our method and without losing generality, let the networks' weights be in the same range as the SA temperature. This is also pretty convenient for monitoring purposes as the current temperature (when cooling or heating) indicates by how much the network's weights are being perturbed.

Once the network's input has been limited and, hence, the network's weights, it is possible to choice an appropriate value for  $k$ . This constant affects the probability of acceptance for a given value of  $\Delta E$ . We found that a value of  $k = 1500$ , for a cooling schedule using an initial temperature of  $15$  and a final temperature of  $0.015$ , provides reasonable learning. Where reasonable learning means that the mse does not wander excessively from high and low values due to an excessive probability of acceptance (read excessive heating). Typical cooling schedules start at a high temperature and gradually cool down using different functions until they reach a specified final temperature, see [5]. On the other hand, the proposed method requires the temperature to increase and decrease periodically. Figure [2].a shows a typical cooling schedule (linear cooling), Figure [2].b shows the exponential cooling schedule which is also very popular, Figure [2].c shows the cooling schedule proposed.

Temperature cycling has been previously used by Möbius et al. to solve the Traveling Salesman Problem, see [9]. On the other hand, to describe how temperature cycling must be used for ANN learning consider the finite length series,  $x[n]$ , defined as

$$x[n + 1] = \rho x[n], \quad n = 1, 2, 3, \dots, N, \quad (2)$$





**Fig. 2.** (a) Exponential cooling, (b) Linear cooling, (b) Temperature cycling

where  $N$  is the number of temperatures,  $x[1]$  is the initial temperature,  $x[N]$  is the final temperature, and  $\rho$  is a cooling constant defined as

$$\rho = e^{\log\left(\frac{(N-1)x[N]}{x[1]}\right)}. \quad (3)$$

The cooling schedule for temperature cycling is defined as

$$y[n] = \sum_{m=0}^{M-1} x[n - mN], \quad (4)$$

where  $y[n]$  is the SA temperature,  $M$  is the number of cycles before starting the optimization process. Additionally, temperature cycling requires keeping the number of iterations at each temperature to a relatively low value, i.e., 10, 20 or 30 are good values. Note that if ANN's training is performed using 100 iterations per temperature or more, the benefit gained for using temperature cycling is lost. Iterating too much at each temperature can be bad for temperature cycling because the solution may fall too much and it will be difficult to escape from this minimum. Additionally, the network's weights must be updated using the recursive equation shown

$$w_{i,j}[n+1] = \gamma(1-\lambda) w_{i,j}[n] + \lambda u[n] - \frac{1}{2}\lambda, \quad (5)$$

where  $w_{i,j}$  is the network's weight connecting the  $j$  neuron with the  $i$  neuron in the next layer,  $\lambda$  is the perturbation ratio defined as

$$\lambda = \frac{y[n]}{x[1]}, \quad (6)$$

$u \in [0, 1)$  is a uniformly distributed random variable, and  $\gamma$  depends on the weights' range. Typical values for this constant are 20 or 30 depending on the network's input. Observe that for a current temperature  $y[n]$  close to the initial temperature  $x[1]$ ,  $\lambda$  takes values close to 1 and the network's weights are violently perturbed. As the temperature decreases the network's weights wandered randomly around a fixed value. Observe also that Equation 5 properly combines the perturbation and the current weight value. Other approaches for SA perturb the network's weights, and then clip the perturbed value to keep the network's weights within a specified range, this is incorrect because some information is lost during the clipping process.

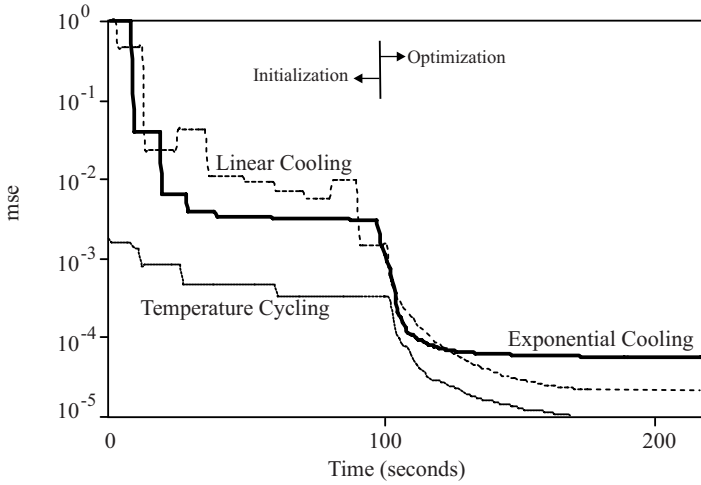
## 4 Simulation Results

For simulation purposes three different multi-layer feed forward neural networks were designed and trained. The first neural network was designed to learn the Trifolium. The resulting auto-associative neural network required 64 inputs, 64 outputs and 4 neurons on the hidden layer. The second neural network was designed to learn the Cardioid, and it required 64 inputs, 64 outputs and 4 neurons on the hidden layer. The third neural network was designed to learn the Lemniscate of Bernoulli, and it required 64 inputs, 64 outputs and 5 neurons on the hidden layer. Because of the number of network's outputs, the network's training may be slow, if the method of Levenberg-Marquardt for optimization is used. Thus, the method of Conjugate-Gradient method was used. It is important to note that the training process was monitored during the initialization and optimization phases separately on an mse-time plot, and this will be explained next.

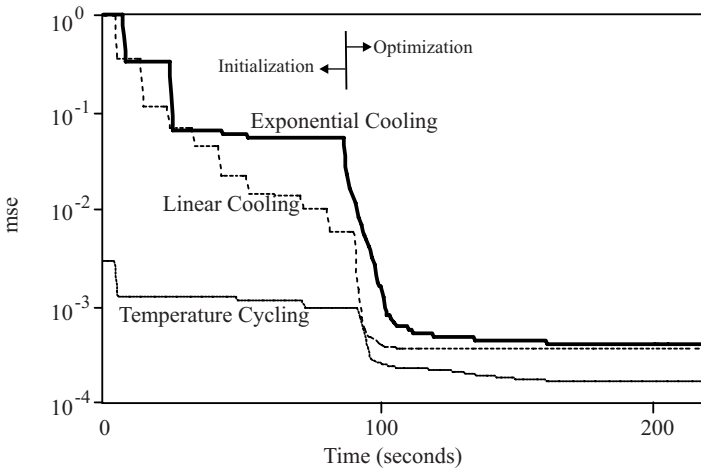
### 4.1 Learning

To illustrate how temperature cycling affects neural network learning. The networks' mse was recorded at equally spaced time intervals during the training process for each of the neural networks designed previously. In Sub-Section 4.2, it will be indicated the number of temperatures, iterations as well as the number of cycles used during the training of each of the three ANNs. Figure 3 shows the network's mse as a function of time for the Trifolium; from this figure, it can be seen that during the initialization phase the method of temperature cycling minimized drastically the mse. Observe that once the initialization phase was completed, the training process switched to the optimization phase. For the exponential and linear cooling schedules the initialization process did not offer a good initial solution, and the optimization process was not able to find the global minimum, and after 200 epochs, they were stuck without hope. On the other hand, the network trained using temperature cycling reached an acceptable mse value during the initialization phase, hence the optimization process attained an mse of  $1 \times 10^{-5}$  by the epoch 180.

Figure 4 and Figure 5 show the networks' mse during the training process for the Cardioid and the Lemniscate of Bernoulli, respectively. As it can be seen

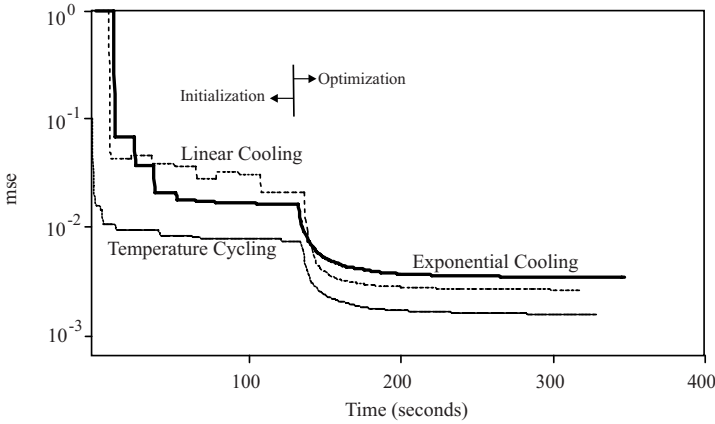


**Fig. 3.** Mean squared error for exponential cooling, linear cooling and temperature cycling while training an auto-associative neural network for learning of the Trifolium



**Fig. 4.** Mean squared error for exponential cooling, linear cooling and temperature cycling while training an auto-associative neural network for learning of the Cardioid

from these figures, the networks trained using temperature cycling outperformed those trained by exponential or linear cooling. Before leaving this section, it is important to mention that the success of temperature cycling requires performing only a few iterations at each temperature. Next, the network’s ability to reduce noise will be discussed and analyzed.

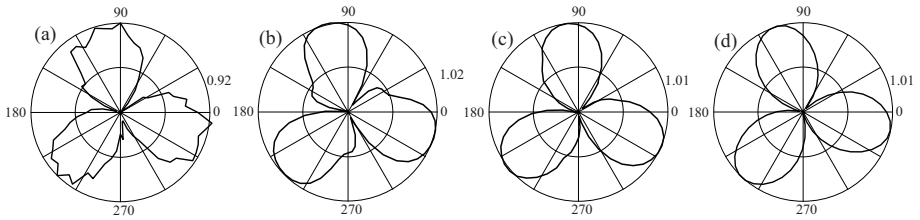


**Fig. 5.** Mean squared error for exponential cooling, linear cooling and temperature cycling while training an auto-associative neural network for learning of the Lemniscate of Bernoulli

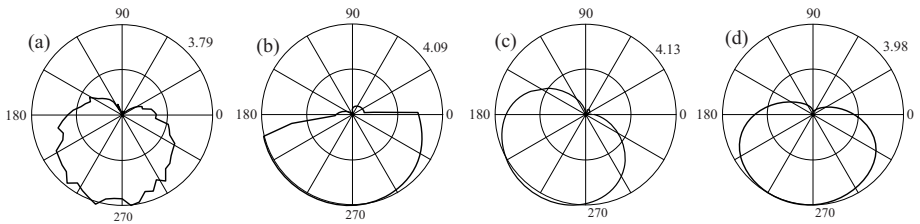
## 4.2 Noise Reduction

Once an appropriate network to learn the Trifolium was designed, it was trained using a set of 780 different training cases; they included the Trifolium shape at several rotation angles using a resolution of 64 points. The number of training cases for this *training set* was computed based on the numbers of network's weights to be adjusted and avoid overfitting, see [10]. Additionally, a Trifolium, with a random rotation angle, was contaminated with noise to test the network and its training. Figure 6.a shows the noisy Trifolium sample. First, an auto-associate neural network was trained using exponential cooling: 10 temperatures and 1000 iterations per temperature. Once the training was completed, the network was excited using the noisy sample of Figure 6.a. The output of this network is shown in Figure 6.b. The same experiment was repeated using linear cooling; Figure 6.c shows the results obtained on this case. Last, another neural network, with the same structure as the used for exponential cooling, was training using temperature cycling: 10 temperatures, 10 iterations per temperature and 100 cooling cycles; note that the same number of iterations was used for all experiments, 10,000 iterations. Figure 6.d shows the output of the neural network trained using temperature cycling; as it can be seen from this figure, the performance of the network trained using temperature cycling is much better than the performance of the neural networks trained using exponential or linear cooling.

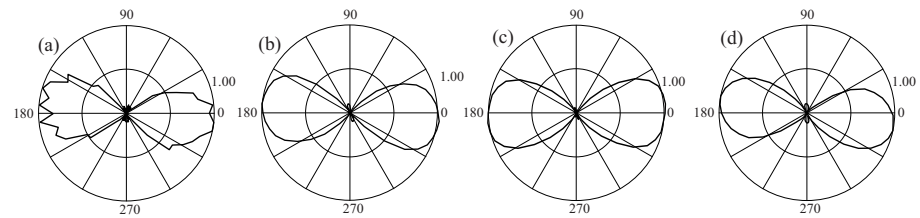
Figure 7.a shows a noisy cardioid. Figure 7.b shows the output of a neural network trained using exponential cooling, Figure 7.c shows the output of a neural network trained using linear cooling and Figure 7.d shows the the output of neural network trained using temperature cycling. Figure 8 shows the results for the Lemniscate of Bernoulli. For all cases, it can be seen that those networks trained using temperature cycling outperform those networks trained using regular cooling for noise reduction.



**Fig. 6.** (a) Noisy Trifolium sample, (b) Noise reduction of a neural network trained using exponential cooling, (c) Noise reduction of a neural network trained using linear cooling, (d) Noise reduction of a neural network trained using temperature cycling



**Fig. 7.** (a) Noisy Cardioid sample, (b) Noise reduction of a neural network trained using exponential cooling, (c) Noise reduction of a neural network trained using linear cooling, (d) Noise reduction of a neural network trained using temperature cycling



**Fig. 8.** (a) Noisy Lemniscate of Bernoulli sample, (b) Noise reduction of a neural network trained using exponential cooling, (c) Noise reduction of a neural network trained using linear cooling, (d) Noise reduction of a neural network trained using temperature cycling

## 5 Summary

There is not yet enough experience with the method of simulated annealing to say definitively what its future place among optimization methods will be. It is not greedy, in the sense that it is not easily fooled by the quick payoff achieved by falling into unfavorable local minima [11].

For auto-associate multi-layer feed forward neural networks (pattern learning) the method of simulated annealing offers great performance when temperature

cycling is used as long as few iterations are use at each temperature. Too many iterations at each temperature prevents the method from continually reducing the mse when training auto-associate neural networks because at each temperature cycle the method might have fallen too much, and it is unable to escape from a false minimum.

Experimental results were obtained using some classical closed curves. The Trifolium, The Cardioid, and The Lemniscate of Bernoulli were used for training three different ANNs. It was shown that temperature cycling reduces both the mse and the training time when compared with exponential cooling. Additionally, these curves were contaminated with noise, then ANN's were used for noise reduction; in general, those networks trained using temperature cycling provided better noise reduction capabilities than those networks trained using exponential or linear cooling.

Classification or generic neural networks do not seem to benefit from temperature cycling when SA is used for initialization. Future work includes the extension of this method (temperature cycling or other cooling schedules) for classification or generic neural network training.

## References

1. Abramson, D., Dang, H., Krishnamoorthy, M.: Simulated Annealing Cooling Schedules for the School Timetabling Problem. *Asia-Pacific Journal of Operational Research* , 1–22 (1999)
2. Huang, M., Romeo, F., Sangiovanni-Vincentelli, A.: An Efficient General Cooling Schedule for Simulated Annealing. In: ICCAD. Proc. of the IEEE International Conf. on Computer Aided Design, pp. 381–384 (1986)
3. Jones, M.T.: AI Application Programming, Charles River Media, 2nd edn. pp. 49–67 (2005)
4. Johnson, D., McGeoch, L.: The Traveling Salesman Problem: A Case Study in Local Optimization. In: Aarts, E.H., Lenstra, J.K. (eds.) *Local Search in Combinatorial Optimization*, Wiley and Sons, Chichester
5. Luke, B.T.: Simulated Annealing Cooling Schedules, (June 1, 2007) available online at <http://members.aol.com/btluke/simanf1.htm>, accessed
6. Masters, T.: *Practical Neural Network Recipes in C++*, pp. 118–134. Academic Press, London (1993)
7. Masters, T.: *Advanced Algorithms for Neural Networks*, pp. 135–156. John Wiley & Sons Inc, Chichester (1995)
8. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, E.: *Journal of Chemical Physics* 21, 1087–1092 (1953)
9. Möbius, A., Neklioudov, A., Díaz-Sánchez, A., Hoffmann, K.H., Fachat, A., Schreiber, M.: Optimization by Thermal Cycling. *Physical Review* 79(22) (1997)
10. Nilsson, N.J.: *Artificial Intelligence: A New Synthesis*, pp. 37–58. Morgan Kaufmann Publishers, San Francisco (1998)
11. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes in C++: The Art of Scientific Computing*, 2nd edn. pp. 448–459. Cambridge University Press, Cambridge (2002)

12. Reed, R.D., Marks II, R.J.: Neural Smothing: Supervised Learning in Feedforward Artificial Neural Networks, pp. 97–112. The MIT Press, Cambridge (1999)
13. Russel, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall, Englewood Cliffs (2002)

# On Conditions for Intermittent Search in Self-organizing Neural Networks

Peter Tiño

University of Birmingham, Birmingham, UK  
p.tino@cs.bham.ac.uk

**Abstract.** Self-organizing neural networks (SONN) driven by softmax weight renormalization are capable of finding high quality solutions of difficult assignment optimization problems. The renormalization is shaped by a temperature parameter - as the system cools down the assignment weights become increasingly crisp. It has been recently observed that there exists a critical temperature setting at which SONN is capable of powerful intermittent search through a multitude of high quality solutions represented as meta-stable states of SONN adaptation dynamics. The critical temperature depends on the problem size. It has been hypothesized that the intermittent search by SONN can occur only at temperatures close to the first (symmetry breaking) bifurcation temperature of the autonomous renormalization dynamics. In this paper we provide a rigorous support for the hypothesis by studying stability types of SONN renormalization equilibria.

## 1 Introduction

There have been several successful applications of neural computation techniques in solving difficult combinatorial optimization problems [1,2]. Self-organizing neural network (SONN) [3] is a neural-based methodology for solving 0-1 assignment problems that has been successfully applied in a wide variety of applications, from assembly line sequencing to frequency assignment in mobile communications.

As usual in self-organizing systems, dynamics of SONN adaptation is driven by a synergy of cooperation and competition. In the competition phase, for each item to be assigned, the best candidate for the assignment is selected and the corresponding assignment weight is increased. In the cooperation phase, the assignment weights of other candidates that were likely to be selected, but were not quite as strong as the selected one, get increased as well, albeit to a lesser degree. The assignment weights need to be positive and sum to 1. Therefore, after each SONN adaptation phase, the assignment weights need to be renormalized back onto the standard simplex e.g. via the softmax function [4,5]. When endowed with a physics-based Boltzmann distribution interpretation, the softmax function contains a temperature parameter  $T > 0$ . As the system cools down, the assignments become increasingly crisp. In the original setting SONN is annealed



so that a single high quality solution to an assignment problem is found. However, it has been reported recently [6] that there exists a critical temperature  $T_*$  at which SONN is capable of powerful intermittent search through a multitude of high quality solutions represented as meta-stable states of SONN adaptation dynamics. It has been hypothesised that the critical temperature may be closely related to the symmetry breaking bifurcation of equilibria in the *autonomous* softmax dynamics.

Unfortunately, at present there is no theory of the SONN adaptation dynamics driven by the softmax renormalization. The first steps towards theoretical underpinning of SONN adaptation driven by softmax renormalization were taken in [7,8,6,9]. For example, [6] suggests to study SONN adaptation dynamics by concentrating on the *autonomous* renormalization process. Indeed, it is this process that underpins the search dynamics in the SONN. Meta-stable states in the SONN intermittent search at the critical temperature  $T_*$  are shaped by stable equilibria of the autonomous renormalization dynamics.

In this paper we rigorously show that the only temperature at which stable renormalization equilibria emerge is the first symmetry breaking bifurcation temperature. The stable equilibria appear close to the corners of the assignment weight simplex (0-1 assignment solutions) and act as pulling devices in the intermittent search towards possible assignment solutions. For a rich intermittent search, the stable equilibria should be only weakly attractive, which corresponds to temperatures lower than, but close to the symmetry breaking bifurcation temperature of the autonomous renormalization dynamics. Due to space limitations, we only provide sketches of proofs of the main statements.

The paper has the following organization: After a brief introduction to SONN and autonomous renormalization in section 2, we introduce necessary background related to renormalization equilibria in section 3. We then study stability types of the renormalization equilibria in section 4, focusing on equilibria close to ‘one-hot’ 0-1 assignment weights in section 4.1. The results are then discussed and summarized in section 5.

## 2 Self-organizing Neural Network and Iterative Softmax

First, we briefly introduce Self-Organizing Neural Network (SONN) endowed with weight renormalization for solving assignment optimization problems (see e.g. [6]). Consider a finite set of input elements (neurons)  $i \in \mathcal{I} = \{1, 2, \dots, M\}$  that need to be assigned to outputs (output neurons)  $j \in \mathcal{J} = \{1, 2, \dots, N\}$ , so that some global cost of an assignment  $\mathcal{A}: \mathcal{I} \rightarrow \mathcal{J}$  is minimized. Partial cost of assigning  $i \in \mathcal{I}$  to  $j \in \mathcal{J}$  is denoted by  $V(i, j)$ . The ‘strength’ of assigning  $i$  to  $j$  is represented by the ‘assignment weight’  $w_{i,j} \in (0, 1)$ .

The SONN algorithm can be summarized as follows: The connection weights  $w_{i,j}$ ,  $i \in \mathcal{I}$ ,  $j \in \mathcal{J}$ , are first initialized to small random values. Then, repeatedly, an output item  $j_c \in \mathcal{J}$  is chosen and the partial costs  $V(i, j_c)$  incurred by assigning all possible input elements  $i \in \mathcal{I}$  to  $j_c$  are calculated in order to select the ‘winner’ input element (neuron)  $i(j_c) \in \mathcal{I}$  that minimizes  $V(i, j_c)$ .

The ‘neighborhood’  $\mathcal{B}_L(i(j_c))$  of size  $L$  of the winner node  $i(j_c)$  consists of  $L$  nodes  $i \neq i(j_c)$  that yield the smallest partial costs  $V(i, j_c)$ . Weights from nodes  $i \in \mathcal{B}_L(i(j_c))$  to  $j_c$  get strengthened:

$$w_{i,j_c} \leftarrow w_{i,j_c} + \eta(i)(1 - w_{i,j_c}), \quad i \in \mathcal{B}_L(i(j_c)), \quad (1)$$

where  $\eta(i)$  is proportional to the quality of assignment  $i \rightarrow j_c$ , as measured by  $V(i, j_c)$ . Weights  $\mathbf{w}_i = (w_{i,1}, w_{i,2}, \dots, w_{i,N})'$  for each input node  $i \in \mathcal{I}$  are then renormalized using softmax

$$w_{i,j} \leftarrow \frac{\exp(\frac{w_{i,j}}{T})}{\sum_{k=1}^N \exp(\frac{w_{i,k}}{T})}, \quad (2)$$

where  $T > 0$  is a ‘temperature’ parameter.

We will refer to SONN for solving an  $(M, N)$ -assignment problem as  $(M, N)$ -SONN. Since meta-stable states in the SONN intermittent search are shaped by stable equilibria of the autonomous renormalization dynamics, in this paper we concentrate on conditions for emergence of such equilibria.

The weight vector  $\mathbf{w}_i$  of each of  $M$  neurons in an  $(M, N)$ -SONN lives in the standard  $(N - 1)$ -simplex,

$$S_{N-1} = \{\mathbf{w} = (w_1, w_2, \dots, w_N)' \in \mathbb{R}^N \mid w_i \geq 0, i = 1, 2, \dots, N, \text{ and } \sum_{i=1}^N w_i = 1\}.$$

Given a value of the temperature parameter  $T > 0$ , the softmax renormalization step in SONN adaptation transforms the weight vector of each unit as follows:

$$\mathbf{w} \mapsto \mathbf{F}(\mathbf{w}; T) = (F_1(\mathbf{w}; T), F_2(\mathbf{w}; T), \dots, F_N(\mathbf{w}; T))', \quad (3)$$

where

$$F_i(\mathbf{w}; T) = \frac{\exp(\frac{w_i}{T})}{Z(\mathbf{w}; T)}, \quad i = 1, 2, \dots, N, \quad (4)$$

and  $Z(\mathbf{w}; T) = \sum_{k=1}^N \exp(\frac{w_k}{T})$  is the normalization factor. Formally,  $\mathbf{F}$  maps  $\mathbb{R}^N$  to  $S_{N-1}^0$ , the interior of  $S_{N-1}$ .

Linearization of  $\mathbf{F}$  around  $\mathbf{w} \in S_{N-1}^0$  is given by the Jacobian  $J(\mathbf{w}; T)$ :

$$J(\mathbf{w}; T)_{i,j} = \frac{1}{T}[\delta_{i,j}F_i(\mathbf{w}; T) - F_i(\mathbf{w}; T)F_j(\mathbf{w}; T)], \quad i, j = 1, 2, \dots, N, \quad (5)$$

where  $\delta_{i,j} = 1$  iff  $i = j$  and  $\delta_{i,j} = 0$  otherwise.

The softmax map  $\mathbf{F}$  induces on  $S_{N-1}^0$  a discrete time dynamics known as *Iterative Softmax* (ISM):

$$\mathbf{w}(t + 1) = \mathbf{F}(\mathbf{w}(t); T). \quad (6)$$

The renormalization step in an  $(M, N)$ -SONN adaptation involves  $M$  separate renormalizations of weight vectors of all of the  $M$  SONN units. For each

<sup>1</sup> Here ‘ $'$  denotes the transpose operator.

temperature setting  $T$ , the structure of equilibria in the  $i$ -th system,  $\mathbf{w}_i(t+1) = \mathbf{F}(\mathbf{w}_i(t); T)$ , gets copied in all the other  $M - 1$  systems. Using this symmetry, it is sufficient to concentrate on a single ISM (6). Note that the weights of different units are coupled by the SONN adaptation step (11). We will study systems for  $N \geq 2$ .

### 3 Renormalization Equilibria

We first introduce basic concepts and notation that will be used throughout the paper. The  $n$ -dimensional column vectors of 1's and 0's are denoted by  $\mathbf{1}_n$  and  $\mathbf{0}_n$ , respectively. The maximum entropy point  $N^{-1}\mathbf{1}_N$  of the standard  $(N - 1)$ -simplex  $S_{N-1}$  will be denoted by  $\bar{\mathbf{w}}$ . To simplify the notation we will use  $\bar{\mathbf{w}}$  to denote both the maximum entropy point of  $S_{N-1}$  and the vector  $\bar{\mathbf{w}} - \mathbf{0}_N$ .

It is obvious that  $\bar{\mathbf{w}}$  is a fixed point of ISM (6) for any temperature setting  $T$ . We will also use the fact (see 9) that any other ISM fixed point  $\mathbf{w} = (w_1, w_2, \dots, w_N)'$  has exactly two different coordinate values:  $w_i \in \{\gamma_1, \gamma_2\}$ , such that  $N^{-1} < \gamma_1 < N_1^{-1}$  and  $0 < \gamma_2 < N^{-1}$ , where  $N_1$  is the number of coordinates  $\gamma_1$  larger than  $N^{-1}$ . The number of coordinates  $\gamma_2$  smaller than  $N^{-1}$  is then  $N_2 = N - N_1$ . Of course, since the assignment weights live on the standard simplex  $S_{N-1}^0$ , we have

$$\gamma_2 = \frac{1 - N_1\gamma_1}{N - N_1}. \tag{7}$$

It is also obvious that if  $\mathbf{w} = (\gamma_1\mathbf{1}'_{N_1}, \gamma_2\mathbf{1}'_{N_2})'$  is a fixed point of ISM (6), so are all  $\binom{N}{N_1}$  distinct permutations of it. We collect  $\mathbf{w}$  and its permutations in a set

$$\mathcal{E}_{N,N_1}(\gamma_1) = \left\{ \mathbf{v} \mid \mathbf{v} \text{ is a permutation of } \left( \gamma_1\mathbf{1}'_{N_1}, \frac{1 - N_1\gamma_1}{N - N_1}\mathbf{1}'_{N-N_1} \right)' \right\}. \tag{8}$$

Finally, fixed points in  $\mathcal{E}_{N,N_1}(\gamma_1)$  exist if and only if the temperature parameter  $T$  is set to 9

$$T_{N,N_1}(\gamma_1) = (N\gamma_1 - 1) \left[ -(N - N_1) \cdot \ln \left( 1 - \frac{N\gamma_1 - 1}{(N - N_1)\gamma_1} \right) \right]^{-1}. \tag{9}$$

### 4 Stability Analysis of Renormalization Equilibria

We have already mentioned that the maximum entropy point  $\bar{\mathbf{w}}$  is a fixed point of ISM (6) for all temperature settings. It is straightforward to show that  $\bar{\mathbf{w}}$ , regarded as a vector  $\bar{\mathbf{w}} - \mathbf{0}_N$ , is an eigenvector of the Jacobian  $J(\mathbf{w}; T)$  at any  $\mathbf{w} \in S_{N-1}^0$ , with associated eigenvalue  $\lambda = 0$ . This simply reflects the fact that ISM renormalization acts on the standard simplex  $S_{N-1}$ , which is a subset of a  $(N - 1)$ -dimensional hyperplane with normal vector  $\mathbf{1}_N$ .

We will now show that if  $\mathbf{w}$  is a fixed point of ISM, then  $\bar{\mathbf{w}} - \mathbf{w}$  is an eigenvector of the ISM Jacobian at  $\mathbf{w}$ .

**Theorem 1.** Let  $\mathbf{w} \in \mathcal{E}_{N,N_1}(\gamma_1)$  be a fixed point of ISM (6). Then,  $\mathbf{w}_* = \bar{\mathbf{w}} - \mathbf{w}$  is an eigenvector of the Jacobian  $J(\mathbf{w}; T_{N,N_1}(\gamma_1))$  with the corresponding eigenvalue  $\lambda_*$ , where

1. if  $\lceil N/2 \rceil \leq N_1 \leq N - 1$ , then  $0 < \lambda_* < 1$ ,
2. if  $1 \leq N_1 < \lceil N/2 \rceil$ ,
  - (a) and  $N^{-1} < \gamma_1 < (2N_1)^{-1}$ , then  $\lambda_* > 1$ .
  - (b) then there exists  $\bar{\gamma}_1 \in ((2N_1)^{-1}, N_1^{-1})$ , such that for all ISM fixed points  $\mathbf{w} \in \mathcal{E}_{N,N_1}(\gamma_1)$  with  $\gamma_1 \in (\bar{\gamma}_1, N_1^{-1})$ ,  $0 < \lambda_* < 1$ .

Sketch of the Proof

To simplify the notation, we will denote the Jacobian of ISM at  $\mathbf{w}=(w_1, w_2, \dots, w_N)'$  by  $J$  and the temperature at which  $\mathbf{w}$  exists by  $T$ . From

$$J\mathbf{w}_* = J(\bar{\mathbf{w}} - \mathbf{w}) = J\bar{\mathbf{w}} - J\mathbf{w} = -J\mathbf{w},$$

and using (5), we have for the  $i$ -th element of  $J\mathbf{w}_*$ :

$$\frac{w_i}{T}(\mathbf{w}'\mathbf{w} - w_i\mathbf{e}'_i\mathbf{w}) = \frac{w_i}{T}(\|\mathbf{w}\|^2 - w_i).$$

But

$$J\mathbf{w}_* = \lambda_*\bar{\mathbf{w}} - \lambda_*\mathbf{w},$$

and so the  $i$ -th element of  $J\mathbf{w}_*$  must also be equal to  $\lambda_*N^{-1} - \lambda_*w_i$ . In other words,

$$\frac{w_i}{T}(\|\mathbf{w}\|^2 - w_i) = \lambda_*N^{-1} - \lambda_*w_i \tag{10}$$

should hold for all  $i = 1, 2, \dots, N$ . Since  $w_i \in \{\gamma_1, \gamma_2\}$ ,  $\gamma_2 = (1 - N_1\gamma_1)/(N - N_1)$ , we have that

$$\frac{\gamma_1 \cdot (\gamma_1 - \|\mathbf{w}\|^2)}{\gamma_2 \cdot (\gamma_2 - \|\mathbf{w}\|^2)} = \frac{\gamma_1 - N^{-1}}{\gamma_2 - N^{-1}} \tag{11}$$

would need to be true. This indeed can be verified after some manipulation using  $\|\mathbf{w}\|^2 = N_1\gamma_1^2 + N_2\gamma_2^2$ .

From (10), by plugging in  $\gamma_1$  for  $w_i$ , we obtain

$$\lambda_* = \frac{\gamma_1 \cdot (\gamma_1 - \|\mathbf{w}\|^2)}{T \cdot (\gamma_1 - N^{-1})}. \tag{12}$$

Now, it can be shown that  $\gamma_1 > \|\mathbf{w}\|^2$ , and since  $\gamma_1 > N^{-1} > 0$ ,  $T > 0$ , we have that  $\lambda_*$  is positive.

It can be established that the values of coordinates  $\gamma_1, \gamma_2$  of each ISM fixed point  $\mathbf{w}$  must satisfy

$$\gamma_1 = \frac{1}{N} \left( 1 + \tau \frac{N_2}{N_1} \right) \tag{13}$$

$$\gamma_2 = \frac{1}{N}(1 - \tau), \tag{14}$$

for some  $\tau \in [0, 1)$ .

In order to prove [1](#), we use parametrization [\(13-14\)](#) to obtain (after some manipulation)

$$\lambda_* = \frac{\gamma_1}{T}(1 - \tau). \tag{15}$$

Assume  $\lambda_* \geq 1$ . That means (using [\(9\)](#), [\(14\)](#), [\(15\)](#), and after some manipulations)

$$\frac{T}{\gamma_1} = \frac{1 - \frac{\gamma_2}{\gamma_1}}{-\ln \frac{\gamma_2}{\gamma_1}} \leq 1 - \tau = \gamma_1 N \frac{\gamma_2}{\gamma_1},$$

which can be written as

$$\ln a \leq -\rho \frac{(1-a)^2}{a} + a - 1 = f(a; \rho), \tag{16}$$

where

$$0 < \rho = \frac{N_1}{N} < 1 \tag{17}$$

and

$$0 < a = \frac{\gamma_2}{\gamma_1} < 1. \tag{18}$$

On  $a > 0$ , both  $\ln a$  and  $f(a)$  are continuous concave functions with  $\ln 1 = f(1) = 0$  and  $\ln'(1) = f'(1) = 1$ . So the function values, as well as the slopes of  $\ln a$  and  $f(a)$  coincide at  $a = 1$ . Since it can be shown that for  $N_1 \geq \lceil N/2 \rceil$  (hence  $\rho \geq 1/2$ ), the slope of  $f(a)$  exceeds that of  $\ln a$  on the whole interval  $(0, 1)$ , it must be that  $f(a) < \ln a$ . This is a contradiction to [\(16\)](#), and so  $\lambda_* < 1$ .

The proof of [2a](#) proceeds analogously to the proof of [1](#), this time we are interested in conditions on  $\gamma_1$  in the neighborhood of  $a = 1$ , such that, given  $\rho \in (0, 1/2)$ , we have  $f'(a) < \ln'(a)$ , and hence  $f(a) > \ln a$ .

Finally, for  $\rho \in (0, 1/2)$ , we have  $f'(a) > \ln'(a)$  for  $a > 0$  in the neighborhood of 0. Now,  $\ln a$  and  $f(a)$  are continuous concave functions with  $\ln 1 = f(1) = 0$ ,  $\ln'(1) = f'(1) = 1$ , and for  $a \in (0, 1)$  in the neighborhood of 1, we have  $f'(a) < \ln'(a)$ , implying  $\ln a < f(a)$ . Since  $\lim_{a \rightarrow 0^+} f(a) = \lim_{a \rightarrow 0^+} \ln a = \infty$  and  $\lim_{a \rightarrow 0^+} \frac{\ln a}{f(a)} = 0$ , there exists  $\bar{a} \in (0, 1)$ , such that on  $a \in (0, \bar{a})$ ,  $f(a) < \ln a$ . It can be shown that  $a \in (0, \bar{a})$  corresponds to  $\gamma_1 \in (\bar{\gamma}_1, N_1^{-1})$  with  $\bar{\gamma}_1 \in ((2N_1)^{-1}, N_1^{-1})$ . This proves [2b](#). Q.E.D.

We have established that for an ISM equilibrium  $\mathbf{w}$ , both  $\bar{\mathbf{w}}$  and  $\mathbf{w}_* = \bar{\mathbf{w}} - \mathbf{w}$  are eigenvectors of the ISM Jacobian at  $\mathbf{w}$ . Stability types of the remaining  $N - 2$  eigendirections are characterized in the next two theorems.

**Theorem 2.** Consider a ISM fixed point  $\mathbf{w} = (\gamma_1 \mathbf{1}'_{N_1}, \gamma_2 \mathbf{1}'_{N_2})'$  for some  $1 \leq N_1 < N$  and  $N^{-1} < \gamma_1 < N_1^{-1}$ . Let  $\mathcal{B} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{N-N_1-1}\}$  be a set of  $(N - N_1)$ -dimensional unit vectors, such that  $\mathcal{B}$ , together with  $\mathbf{1}_{N-N_1} / \|\mathbf{1}_{N-N_1}\|$ , form an orthonormal basis of  $\mathbb{R}^{N-N_1}$ . Then, there are  $N - N_1 - 1$  eigenvectors of the ISM Jacobian at  $\mathbf{w}$  of the form:

$$\mathbf{v}_-^i = (\mathbf{0}'_{N_1}, \mathbf{u}_i)'. \quad i = 1, 2, \dots, N - N_1 - 1. \tag{19}$$

All eigenvectors  $\mathbf{v}_-^1, \mathbf{v}_-^2, \dots, \mathbf{v}_-^{N-N_1-1}$  have the same associated eigenvalue

$$0 < \lambda_- = \frac{1 - N_1\gamma_1}{(N - N_1)T_{N,N_1}(\gamma_1)} = \frac{\gamma_2}{T_{N,N_1}(\gamma_1)} < 1. \tag{20}$$

Sketch of the Proof

The Jacobian can be written as

$$J = \frac{-1}{T} \begin{bmatrix} G_1 & G_{12} \\ G'_{12} & G_2 \end{bmatrix}, \tag{21}$$

where

$$G_1 = \gamma_1(\gamma_1 \mathbf{1}_{N_1} \mathbf{1}'_{N_1} - I_{N_1}), \tag{22}$$

$$G_2 = \gamma_2(\gamma_2 \mathbf{1}_{N_2} \mathbf{1}'_{N_2} - I_{N_2}), \tag{23}$$

and

$$G_{12} = \gamma_1 \gamma_2 \mathbf{1}_{N_1} \mathbf{1}'_{N_2}. \tag{24}$$

Since all  $\mathbf{u} \in \mathcal{B}$  are orthogonal to  $\mathbf{1}_{N_2}$ , we have

$$\begin{bmatrix} G_{12} \\ G_2 \end{bmatrix} \mathbf{u} = -\gamma_2 \begin{bmatrix} \mathbf{0}_{N_1} \\ \mathbf{u} \end{bmatrix}.$$

and so for all  $i = 1, 2, \dots, N - N_1 - 1$ , it holds

$$J\mathbf{v}_-^i = \frac{\gamma_2}{T} \mathbf{v}_-^i.$$

Since both  $\gamma_2$  and  $T$  are positive,  $\lambda_- = \gamma_2/T > 0$ .

It can be shown that

$$T = \frac{\gamma_1 - \gamma_2}{\ln \gamma_1 - \ln \gamma_2}. \tag{25}$$

Then,

$$\lambda_- = \frac{\ln \frac{\gamma_1}{\gamma_2}}{\frac{\gamma_1}{\gamma_2} - 1} = \frac{\ln b}{b - 1}, \tag{26}$$

where  $b = \gamma_1/\gamma_2 > 1$ . But  $0 < \ln b < b - 1$  on  $b \in (1, \infty)$ , and so  $\lambda_- < 1$ . *Q.E.D.*

Note that even though theorem 2 is formulated for  $\mathbf{w} = (\gamma_1 \mathbf{1}'_{N_1}, \gamma_2 \mathbf{1}'_{N_2})'$ , by the symmetry of ISM, the result translates to all permutations of  $\mathbf{w}$  in a straightforward manner. The same applies to the next theorem.

**Theorem 3.** Consider a ISM fixed point  $\mathbf{w} = (\gamma_1 \mathbf{1}'_{N_1}, \gamma_2 \mathbf{1}'_{N_2})'$  for some  $1 \leq N_1 < N$  and  $N^{-1} < \gamma_1 < N_1^{-1}$ . Let  $\mathcal{B} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{N_1-1}\}$  be a set of  $N_1$ -dimensional unit vectors, such that  $\mathcal{B}$ , together with  $\mathbf{1}_{N_1}/\|\mathbf{1}_{N_1}\|$ , form an orthonormal basis of  $\mathbb{R}^{N_1}$ . Then, there are  $N_1 - 1$  eigenvectors of the ISM Jacobian at  $\mathbf{w}$  of the form:

$$\mathbf{v}_+^i = (\mathbf{u}'_i, \mathbf{0}'_{N-N_1})', \quad i = 1, 2, \dots, N_1 - 1. \tag{27}$$

All eigenvectors  $\mathbf{v}_+^1, \mathbf{v}_+^2, \dots, \mathbf{v}_+^{N_1-1}$  have the same associated eigenvalue

$$\lambda_+ = \frac{\gamma_1}{T_{N,N_1}(\gamma_1)} > 1. \tag{28}$$

Sketch of the Proof

The proof proceeds analogously to the proof of theorem 2 Q.E.D.

**4.1 Equilibria Near Vertices of the Standard Simplex**

We have proved that components of *stable* equilibria of the SONN renormalization can only be found when  $N_1 = 1$ , which corresponds to equilibria near vertices of the simplex  $S_{N-1}$ . Stable manifold of the linearized ISM system at such an equilibrium  $\mathbf{w} \in S_{N-1}^0$  is given by the span of  $\mathbf{w}_* = \bar{\mathbf{w}} - \mathbf{w}$  (by theorem 1 the corresponding eigenvalue  $\lambda_*$  is in  $(0, 1)$  if  $\mathbf{w}$  lies sufficiently close to a vertex of  $S_{N-1}$ ) and  $N - 2$  vectors orthogonal to both  $\bar{\mathbf{w}}$  and  $\mathbf{w}_*$  (theorem 2). Since  $N_1 = 1$ , there is no invariant manifold of the linearized ISM with expansion rate  $\lambda_+ > 1$  (theorem 3). Also, no dynamics takes place outside interior of  $S_{N-1}$  (zero eigenvalue corresponding to the eigenvector  $\bar{\mathbf{w}}$  of the linearized ISM). It is important to note that by theorem 3, *all* ISM fixed points with  $N_1 \geq 2$  are unstable. From now on we will concentrate on ISM equilibria with  $N_1 = 1$ .

The temperature  $T_{N,1}(\gamma_1)$  (see eq. (9)) at which fixed points  $\mathbf{w} \in \mathcal{E}_{N,1}(\gamma_1)$  exist is a concave function attaining a unique maximum at some  $\gamma_1^0 \in (N^{-1}, 1)$  (9). Denote the corresponding temperature  $T_{N,1}(\gamma_1^0)$  by  $T_E(N, 1)$ . Then, no fixed point  $\mathbf{w} \in \mathcal{E}_{N,1}(\gamma_1)$  for any  $\gamma_1 \in (N^{-1}, 1)$  can exist for temperatures  $T > T_E(N, 1)$ . For temperatures  $T$  slightly below  $T_E(N, 1)$  there are two fixed points  $\mathbf{w}_-(T)$  and  $\mathbf{w}_+(T)$ , corresponding to the increasing and decreasing branches of the concave temperature curve  $T_{N,1}(\gamma_1)$ , respectively. Temperature  $T_E(N, 1)$  is the first symmetry breaking bifurcation temperature of the ISM when new fixed points other than  $\bar{\mathbf{w}}$  emerge as the system cools down.

In this section we show that one of the fixed points,  $\mathbf{w}_+(T)$ , is a stable equilibrium with increasingly strong attraction rate as the system cools down ( $\mathbf{w}_+(T)$  moves towards a vertex of  $S_{N-1}$ ), while the other one,  $\mathbf{w}_-(T)$ , is a saddle ISM fixed point.

By (12) (and some manipulation),  $\bar{\mathbf{w}} - \mathbf{w}$  is an eigenvector of the ISM Jacobian at  $\mathbf{w}$  with the corresponding positive eigenvalue

$$\lambda_*(\gamma_1) = \frac{N\gamma_1(1 - \gamma_1)}{N\gamma_1 - 1} \ln \frac{(N - 1)\gamma_1}{1 - \gamma_1}. \tag{29}$$

It is straightforward to show that on  $\gamma_1 \in (N^{-1}, 1)$ ,  $\lambda_*(\gamma_1)$  is a concave function of positive slope at  $\gamma_1 = N^{-1}$  with

$$\lim_{\gamma_1 \rightarrow N^{-1}} \lambda_*(\gamma_1) = 1$$

and

$$\lim_{\gamma_1 \rightarrow 1} \lambda_*(\gamma_1) = 0.$$

Hence there exists a unique  $\gamma_1^* \in (N^{-1}, 1)$ , such that  $\lambda_*(\gamma_1^*) = 1$  and for  $\gamma_1 \in (N^{-1}, \gamma_1^*)$  we have  $\lambda_*(\gamma_1) > 1$ , whereas for  $\gamma_1 \in (\gamma_1^*, 1)$ , it holds  $0 < \lambda_*(\gamma_1) < 1$ .

---

<sup>2</sup>  $\lambda_*(\gamma_1)$  can be continuously extended to  $\gamma_1 = 1/N$ .

From (29) and  $\lambda_*(\gamma_1^*) = 1$ , we have

$$\ln \frac{(N-1)\gamma_1^*}{1-\gamma_1^*} = \frac{N\gamma_1^* - 1}{N\gamma_1^*(1-\gamma_1^*)}. \quad (30)$$

It is not difficult to show that  $\gamma_1^*$  is actually equal to  $\gamma_1^0$ , the value of  $\gamma_1$  at which  $T_{N,1}(\gamma_1)$  attains maximum. As the system gets annealed, the eigenvalue  $\lambda_*(\gamma_1)$  decreases below 1 and increases above 1 for the two fixed points  $\mathbf{w}_+(T)$  and  $\mathbf{w}_-(T)$ , respectively. The weakest contraction is at equilibria  $\mathbf{w}_+(T)$  existing at temperatures close to  $T_E(N, 1)$ .

It can be easily shown that the other contraction rate (theorem 2, (20)),

$$\lambda_-(\gamma_1) = \frac{1-\gamma_1}{(N-1)T_{N,1}(\gamma_1)}, \quad (31)$$

is a decreasing function of  $\gamma_1$  as well. Hence,  $\mathbf{w}_+(T)$  are *stable* equilibria with weakest contraction for temperatures close to  $T_E(N, 1)$ . As the system cools down the contraction gets increasingly strong.

## 5 Discussion – SONN Adaptation Dynamics

In the intermittent search regime by SONN, the search is driven by pulling promising solutions temporarily to the vicinity of the 0-1 ‘one-hot’ assignment values - vertices of  $S_{N-1}$  [6]. The critical temperature for intermittent search should correspond to the case where the attractive forces already exist in the form of *attractive* equilibria near the ‘one-hot’ assignment suggestions (vertices of  $S_{N-1}$ ), but the convergence rates towards such equilibria should be sufficiently weak so that the intermittent character of the search is not destroyed. In this paper, we have rigorously shown that this occurs at temperatures lower than, but close to the first bifurcation temperature  $T_E(N, 1)$ .

The hypothesis that there is a strong link between the critical temperature for intermittent search by SONN and bifurcation temperatures of the autonomous ISM has been formulated in [6]. Both [9] and [6] hypothesize that even though there are many potential ISM equilibria, the critical bifurcation points are related only to equilibria near the vertices of  $S_{N-1}$ , as only those can be ultimately responsible for 0-1 assignment solution suggestions in the course of intermittent search by SONN. In this study, we have rigorously shown that the stable equilibria can in fact exist *only* near the vertices of  $S_{N-1}$ . Only when  $N_1 = 1$ , there are no expansive eigendirections of the local Jacobian with  $\lambda_+ > 1$ .

Even though the present study helps to shed more light on the prominent role of the first (symmetry breaking) bifurcation temperature  $T_E(N, 1)$  in obtaining the SONN intermittent search regime, many interesting open questions remain. For example, no theory as yet exists of the role of abstract neighborhood  $\mathcal{B}_L(i(j_c))$  of the winner node  $i(j_c)$  in the cooperative phase of SONN adaptation. [6] report a rather strong pattern of increasing neighborhood size with increasing assignment problem size (tested on N-queens), but this issue deserves a more detailed study.



We conclude by noting that it may be possible to apply the theory of ISM in other assignment optimization systems that incorporate the softmax assignment weight renormalization e.g. [10,11].

## References

1. Hopfield, J., Tank, D.: Neural computation of decisions in optimization problems. *Biological Cybernetics* 52, 141–152 (1985)
2. Smith, K.: Neural networks for combinatorial optimization: a review of more than a decade of research. *INFORMS Journal on Computing* 11, 15–34 (1999)
3. Smith, K., Palaniswami, M., Krishnamoorthy, M.: Neural techniques for combinatorial optimization with applications. *IEEE Transactions on Neural Networks* 9, 1301–1318 (1998)
4. Guerrero, F., Lozano, S., Smith, K., Canca, D., Kwok, T.: Manufacturing cell formation using a new self-organizing neural network. *Computers & Industrial Engineering* 42, 377–382 (2002)
5. Kosowsky, J., Yuille, A.: The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural Networks* 7, 477–490 (1994)
6. Kwok, T., Smith, K.: Optimization via intermittency with a self-organizing neural network. *Neural Computation* 17, 2454–2481 (2005)
7. Kwok, T., Smith, K.: Performance-enhancing bifurcations in a self-organising neural network. In: Mira, J.M., Álvarez, J.R. (eds.) *IWANN 2003*. LNCS, vol. 2686, pp. 390–397. Springer, Heidelberg (2003)
8. Kwok, T., Smith, K.: A noisy self-organizing neural network with bifurcation dynamics for combinatorial optimization. *IEEE Transactions on Neural Networks* 15, 84–88 (2004)
9. Tiño, P.: Equilibria of iterative softmax and critical temperatures for intermittent search in self-organizing neural networks. *Neural Computation* 19, 1056–1081 (2007)
10. Gold, S., Rangarajan, A.: Softmax to softassign: Neural network algorithms for combinatorial optimization. *Journal of Artificial Neural Networks* 2, 381–399 (1996)
11. Rangarajan, A.: Self-annealing and self-annihilation: unifying deterministic annealing and relaxation labeling. *Pattern Recognition* 33, 635–649 (2000)

# Similarity Clustering of Music Files According to User Preference

Bastian Tenbergen

Human-Computer Interaction M.A. Program  
State University of New York at Oswego  
Oswego, NY, 13126

Formerly:  
Cognitive Science Bachelor Program  
School of Human Sciences  
University of Osnabrück  
Germany

**Abstract.** A plug-in for the Machine Learning Environment Yale has been developed that automatically structures digital music corpora into similarity clusters using a SOM on the basis of features that are extracted from files in a test corpus. Perceptually similar music files are represented in the same cluster. A human user was asked to rate music files according to their subjective similarity. Compared to the user's judgment, the system had a mean accuracy of 65.7%. The accuracy of the framework increases with the size of the music corpus to a maximum of 75%. The study at hand shows that it is possible to categorize music files into similarity clusters by taking solely mathematical features into account that have been extracted from the files themselves. This allows for a variety of different applications like lowering the search space in manual music comparison, or content-based music recommendation.

## 1 Introduction

During the last few years, the channels of acquiring musical data have increased rapidly. Due to the growing importance of broad-band network connections, users are no longer limited to radio or television broadcasting services or other non-computer aided methods to acquire copies of musical pieces. More or less legal peer-to-peer networks, online music stores and personalized Internet radio stations provide the user with a constant and never-ending flow of musical information. On the one hand this large supply makes it possible to get any piece of music any time with a minimal amount of costs. On the other hand, this gives rise to a new, challenging problem: Structuring the music collection of an individual user or a user community to allow for tasks like content-based music recommendation. Keeping the overview over a large collection becomes more difficult and costly with the size of the collection. Unnamed files or files with an unknown content are very difficult to identify and classify manually, which makes a system desirable that is able to accomplish the task

of disambiguating the information about a collection automatically, so two or more users can share these disambiguated information.

Many different research teams have addressed the problem of music information retrieval in different ways. Research is being conducted on the basis of non-real-world audio data, [13], [16], [17], i.e. data of musical information that need to be interpreted in one form or another. Among this kind of musical information belong both score representations of pieces as well as music stored as MIDI files. Although research in musical information retrieval based on non-real-world data sets reveals very promising results, the problem is that those kinds of data sets are usually not comparable to the collections that are stored on a user's hard disc or on broadcasting station servers. It is more likely that a form of compressed or uncompressed real-world audio data (like compact discs or MP3 files) can be found in a collector's database (throughout this work, the term "user" will be used to refer to any kind of individuals that possess and/or consume audio data – no matter if it is a home user, a radio or television station, a web-radio service or a music store, either online or not). Because of this, a system that can deal with real-world audio data is of a much higher value for single users or user groups than a system that can deal with pre-interpreted score- or MIDI-like representations instead. Processing real-world audio data for music information retrieval purposes can generally be done using two prominent approaches: a statistical or mathematical approach [10] and a more neurophysiological approach [15], both possibly involving neural networks [14]. Since the ability to retrieve musical information from real-world audio data is a very impressive capability of the human brain, especially the application of artificial neural networks provides distinguished results. Both approaches in dealing with real-world data make it necessary to extract certain audio features from the data before hand.

Pardo, Shlfrin and Birmingham [13] have conducted a pilot study in matching a sung query to a database of stored MIDI files. They especially address the problem of erroneous queries due to a low humming performance of the user. The outcomes of the experiments were that specifically designed user-based error models provide much better results in handling a mismatch between query and target songs than the complete absence of any error model. Yet, synthetic error models provide equally good results than data-driven, singer-based error models. However, in order to provide a framework for music recommendation, relying on the users to sing or hum a query seems not very applicable. Hence, it might be more useful to have a system that collects and processes data from the collection itself.

Tzanetakis et al. [15] have contributed an important milestone in musical genre classification. The authors argue that human listeners can classify musical content by referring only to small excerpts of a piece, e.g. a few fractions of a second. Hence, human listeners must rely on musical surface features instead of complex (mathematical) interaction between waveforms of the piece. So, Tzanetakis et al. focus on the extraction of surface and rhythm properties and propose a set of own features. The authors proposed two different graphical user interfaces (GUIs) that visualize a song's membership to a certain genre. Since the borderlines of musical genres are rather fuzzy, a nice side effect of the GUIs is that not only the genre of a song is displayed but rather a number of other genres the song shows characteristics of are visualized.

Up to this point, all researchers who extract features from digital audio data mostly presented their own features. This leads to a large variety of features that can be extracted from audio data. However, a unifying framework that measures the features' applicability was missing. Mierswa and Morik [10] were the first researchers to address this problem. In order to accomplish this task, Mierswa and Morik made use of a Genetic Programming approach to explore the space of all possible feature extraction methods in order to learn the feature extraction method that has an ideal performance to classify audio data into genres. Three test corpora were compiled and consisted of music files that belong to either the Pop or Techno genre, the Pop or Hip hop genre or to Pop or Classic music. The files in every corpus needed to be classified in either of the two genres in the corpus.

The article at hand aims at a more general approach to the problem of music collection structuring, as most users probably do not have a pre-structured database. A system is suggested that assists users in the task of finding music pieces by presenting clusters of similar songs according to an input query. This way, a possibly very large collection of song files can be narrowed down to a choice of much fewer songs. This allows for lowering the complexity in comparing songs manually, since a global comparison has already been performed by the system. To accomplish this task, a plug-in for the Machine Learning Environment Yale has been developed. The plug-in makes use of a Self-Organizing Map [4], [5] to cluster audio files and to create similarity clusters. The files need to be processed in a feature extraction algorithm that extracts certain features [10] from the underlying corpus.

Dividing songs into similarity clusters allows for a variety of different tasks. As an example, meta information for unnamed song files can be retrieved. Unidentified files can easily be compared with songs in a cluster containing much fewer elements than the original corpus (which might be excessively large). This helps user groups as well as single users to keep an overview of large music databases. In addition, the suggested system can help automating user preference analysis to individualize the offer of online music stores and other personalized music services.

## 2 Method

### 2.1 Experimental Set-Up

In order to classify musical data into perceptually similar clusters, a plug-in for the Machine Learning Environment Yale has been developed. The plug-in contains a simple Self-Organizing Map operator. This Self-Organizing Map basically lays a network of artificial neurons over the feature vectors. Each feature vector represents a music file in the corpus that needs to be clustered and consists of a number of generic features (like mean loudness, maximum pitch, average beats per minute, etc. [10]). Essentially, the SOM performs a mapping of the high dimensionality of the feature vectors onto a two dimensional representation. The result is that the SOM's layout changes according to the distribution of the feature vectors so that each pattern that naturally exists in the vector space is represented by one or more neurons.

The features are extracted from the music files by making use of a feature extraction method that has provided good results on a variety of different tasks in other research [10].

The underlying requirement of all experiments is that the audio corpus on which the experiments are performed on consists of generic MPEG-1 Layer III files with a sampling rate of 44.1KHz [3]. The proposed clustering framework is robust against corpora consisting of files encoded with different bitrates.

The primary test corpus consisted of 360 audio files, all differing in length, bitrate and genre. These files were subdivided into six sub-corpora: three small ones, each containing approximately 20 files, and three medium ones, each containing 100 files. The files in the small sub-corpora were taken from a private collection and vary in style and genre. The medium corpora contained the (unofficial) top 100 chart songs from the years 1990, 2000 and 2005, respectively. To test the proposed system for applicability in very large collections, a secondary corpus consisting of 1,554 files, i.e. the (unofficial) charts from the years 1990 to 2005 has been compiled and processed. Although an effort has been made to take songs from as many different genres as possible, this research focuses on modern popular music, ranging from early pop stages to fairly recent modern publications. Recapitulating, the overall music corpus consisted of approximately 8.07 gigabytes in 1,914 files summing up to over 160 hours of total playtime. The files in the source directory are generally not structured although recursive processing of subdirectories is supported to take databases into account that have been pre-structured by a user. Similarity clusters are found on the bases of a query file. This file is located within the source directory and is hence processed in terms of feature extraction too. The query file represents a file, the user is interested in finding similar songs to.

## 2.2 Procedure

The basic experimental procedure has been divided into two steps. In the first step, the raw data are transformed into a feature vector representation by simply extracting the features from the music files in the test corpus. The output of the feature extraction step is a file containing the feature vectors. In the second step, this file is used by the clustering SOM operator to train the Self-Organizing Map. In addition, the SOM operator retrieves the file name of the query file from the raw audio data. After the clustering process is done, this file name is retrieved from the similarity clusters that have emerged from the process. The system's output is a string representation of all file names of the songs that are in the same cluster as the query file.

In order to test the clustering performance, the Self-Organizing Map was applied on the extracted features with a variety of different parameter settings for the overall number of neurons. Since the number of clusters cannot exceed the number of neurons in the SOM, but increasing the number of neurons negatively influences the time needed to complete the clustering, an ideal number of neurons that allows for a maximum of clusters needs to be found – a trade-off. All test runs of the SOM had the following parameters in common. The network topology was always set to an “open“

$$n \times m \tag{1}$$

matrix, with

$$n, m > 1, \quad (2)$$

i.e. the neurons in the SOM were aligned in a rectangular structure – the first and last neuron in every row and column were not considered to be neighbors. The number of training runs was set to 1,000 and the learning rates for the winner neuron and its neighboring neurons was set to 1 and 0.25 respectively. According to [14], no superiority of any distance metric over another metric exists. Hence, all test runs have been performed using the Euclidean Distance as the metric to measure the distance of every neuron on the SOM to every input feature vector (that represents a certain song).

To test, which features are most important for the classification task, test runs were performed with normalized and filtered patterns. The patterns have been normalized by dividing each pattern  $x_{i,j}$  – with  $i$  being the  $i$ -th pattern (i.e.  $i$ -th song from the source directory) and  $j$  being the  $j$ -th feature of the  $i$ -th song – by the root mean square deviation of the respective column in the pattern matrix. A filtering of the patterns was done by setting every pattern with a very small absolute value, i.e.

$$x_{i,j} \leq 1, \quad (3)$$

to zero.

From each corpus, one file was chosen to be the query file. Since the query file is the file that users will enter as a sample of their musical taste (for instance), the evaluation focuses on the cluster containing this file to ensure that the suggested similar songs are indeed similar.

To evaluate the similarity of the songs in the cluster containing the query file, the cluster was transposed into a play list and evaluated by a human user. The user was asked to rate each song on a 6-point scale (0: no similarity; 1: little similarity; 2: medium to little similarity; 3: medium to high similarity; 4: high similarity; 5: very similar/close to equal) using the query file as the prototype. The user was asked to consider two songs  $a$  and  $b$  similar, if the

musical style of both songs is subjectively similar,  
musical themes can be described as subjectively similar, and  
pace and/or rhythms of the songs are subjectively similar.

Two songs can also be considered similar, if they are not members of the same (subjective) genre. The user was a twenty-two year old male German student, with no background in music theory and music science. The participation in this study was on a voluntary basis and the user did not receive a reward. Prior the ranking of the songs, the user was naive to the purpose of this study to avoid a user bias. The songs were presented to the user in a random order. Before the user evaluated each song in the similarity cluster, the prototype song was presented to him. He could listen to the song as often as he wanted throughout the evaluation phase.

### 3 Results

While conducting the experiments, special attention has been paid to the performance of the clustering process of the Self-Organizing Map. Generally speaking, the

Self-Organizing Map was able to find similarity clusters among the music files and hence provided good results in structuring the source directory.

In the small test corpora, only a very limited number of clusters could be found. Increasing the number of neurons did not influence the number of clusters. Accordingly, the optimal number of neurons for the small and medium test corpora does not depend on the number of found similarity clusters as well, as long as the SOM contained more neurons than the maximum number of clusters that can be found with respect to a test corpus (that is in worst case: one cluster per song in the corpus).

For every test run, an average of 12 clusters could be found. Optimal clustering performance was achieved by setting the number of neurons to approximately 30% of the number of files in the test corpus. As it was expectable, a very large number of clusters was found in the large test corpus. However, the number of found clusters did not exceed 25% of the number of files in the corpus. For no test run, a very large number of neurons (i.e. in cases in which the number of neurons exceeded the number of files in the corpus by a multiple) influenced the clustering performance negatively.

Normalizing the feature patterns negatively influenced the clustering performance. In all test runs, only two clusters could be found – independent from the chosen target file. Optimal results were provided in test runs in which normalizing of the patterns did not take place. It is to note that the extracted feature vectors have a very wide scope, i.e. some features have a very small absolute value, while others have a very large absolute value. Filtering the feature patterns in every test run did not influence the clustering performance negatively. In most cases, the clusters found with the filtering option activated were the same as the similarity clusters without filtering having taken place.

Generally speaking, applying the SOM on the features extracted from the small corpora resulted in two or three clusters that are found amongst the input files. Although filtering did not influence any of the other test corpora, in the first small corpus, finding similar files was highly dependent on whether the feature patterns were filtered or not. Without filtering, the determined clusters only contained two files, both of which were rated to be very not similar to the target song (target song: “Paddy’s Sicknote“ by “The Dubliners“, Songs in cluster: Marilyn Manson – “Tainted Love” and Frank Boeijen Groep – “Kronenburg Park”). Filtering the patterns caused 14 songs to be in the similarity cluster of the target song, none of which were rated to be similar. The user judged the similarity of “Paddy’s Sicknote” and it became obvious that only “Yellow Submarine” by “The Beatles” was considered to be slightly similar to the target song (it received a ranking of one).

Repeating the test runs on this small corpus with another query file did not reproduce these results. Instead, the results were comparable to the results of test runs on the other corpora.

The clustering process provided good results in structuring the input space into similarity clusters. In all test corpora, similarity clusters could be determined with regard to a specific target file. With regard to the medium corpora and the large corpus, 23.7 similar files could be found in each cluster in average. These files were

rated by a user to judge the subjective similarity between the files. The majority of the files in the similarity clusters were considered to be similar to the target file. In each similarity cluster, there was an average of 14.7 similar files. These files achieved at least a ranking of three on the 6-point scale. Hence, in average, 65.7% of the files in the retrieved similarity clusters are considered to be similar to the query file. Only a few files in the found similarity clusters were not considered similar by the user (average: 9 files, that is 34.3%); these files received a ranking of two or less.

More interestingly, most of the songs that were not enclosed in the similarity cluster of the query file (i.e. those songs that the system considers not similar to the query file) were not considered similar by the human user, too. Only 17% received a ranking of three or above and can be hence thought of as “forgotten” by the system.

Another important finding is the fact that the accuracy of the clustering process was very high in the large corpus. For instance, there were 24 songs in the similarity cluster for the song “Nothing else Matters” by Metallica, 75% of which received a ranking of at least three (18 files), and 54% received a ranking of four or higher (13 files). However, it is to note that a lot of songs that were present in this song's similarity cluster during other test runs (on one of the medium test corpora) were not included in the similarity cluster in the test run on the large corpus.

## 4 Discussion

Although the clustering accuracy of the suggested system is not as high as initially desired, clustering music files into similarity clusters is a good way to structure the search space (i.e. a very large music corpus). The suggested system presents itself as a capable tool to help structuring large musical databases to find music pieces according to the taste of a user or customer, or just structure the database. Nevertheless, there are a few things to note about the results from the previous section.

Filtering the feature patterns in every test run did generally not influence the clustering performance, which proves that features with a very small absolute value do not influence a classification and only play a minor role in the clustering process. These features can hence be omitted and do not need to be extracted from the source files. Therefore, the generic feature set that has been used to extract features from the songs can be designed more stream-lined, i.e. specifically for the task of feature extraction for similarity clustering using a Self-Organizing Map.

The clustering performance is directly connected to the amount of files in the underlying directory. The more files are in the source directory, the higher the clustering performance, regarding the number of retrieved clusters. This, of course, is not surprising since common sense suggests that the higher the number of songs that are included, the higher the diversity among those songs will be. This explains the poor clustering performance when using “Paddy’s Sicknote” as the prototype song, as explained in the previous section. Since this song is a rather unusual one (Irish Gaelic Folk), it is unlikely that a similar song is found in a small cluster. In addition, it is to



note that the number of files falsely included into the similarity cluster of a target song is much lower in larger corpora than in smaller corpora. This can be explained with the fact that it is more likely to find truly similar files with regard to a certain target song in a larger corpus than in a smaller one. In the smaller corpus, the matches that are just “relatively” similar are included because these files are more similar to the target song than to any other song in the corpus – there are just no pieces that are more similar to the target song. However, much more similar songs might be present in the larger corpus. This also explains, why music pieces that were considered similar in a smaller cluster and hence included in a similarity cluster of a certain song are not included in the similarity cluster of the same song in a larger corpus (see the example of the song “Nothing else Matters” in the previous section). Songs, previously considered similar are simply more similar to another song than to the target song in the environment of a larger corpus.

Recapitulating it can be said that the very challenging task to structure large music databases in perceptually similar clusters and hence helping the user to find similar sounding songs more easily can be accomplished by the system at hand. The Self-Organizing Map plug-in for the Machine Learning Environment Yale provided good results in finding similarity clusters on different test corpora and is a competent framework to structure digital music databases automatically so user interaction on the same database is simplified as a database structuring disambiguates the data. Hence, the system can not only be used to help a user finding more music according to his/her personal taste, but it can be used by owners of very large music libraries to manage their databases. This might even be possible “on-the-fly” by extracting features of songs that are added to the library and feeding them into the clustering operator that has already been trained on the rest of the database. The suggested framework can also help in designing offers for customers of music online stores or Internet radio services. It is also imaginable that music pieces can be compared on a topological level, for example by music historians, to visualize musical correlation and similarity.

This can help to answer questions of plagiarism with regard to melody similarity.

Since the random sample of the user judgment was very small in this study, more thorough user surveys need to be conducted with regard to the similarity judgment in successive research to answer the question if the presented system is applicable in wider user communities.

Future work will also address the design of a more streamlined feature extraction method to provide a feature set that allows for more accurate clustering tasks. In addition, the presented system can be tested on a larger variety of music files. The system has mainly been tested on fairly modern and recent music but should be tested on older music and/or classic music. Future research should answer the question of the importance of corpus pre-structuring as well as genre distribution among the test files.

Another interesting future task is to develop a standalone system out of the present plug-in. A piece of software is desirable that works independently from Yale and is

able to provide home users or work groups with an intelligent structuring of their music archives. It is imaginable to include the possibility to share extracted features and store them in a centralized server, which will provide the opportunity to search remote databases for similar songs without violating copyright laws. Connecting online music stores with that database can make it possible for the users to quickly obtain legal copies of music files that are similar to one's private collection for a minimum of financial costs.

## References

1. Allamanche, E., Herre, J., Hellmuth, O., Fröba, B., Kastner, T., Cremer, M.: Content-based identification of audio material using MPEG-7 low level description. In: ISMIR. Proceedings of the Second Annual International Symposium on Music Information Retrieval, pp. 197–204 (2001)
2. Fischer, S., Klinkenberg, R., Mierswa, I., Ritthoff, O.: YALE: Yet Another Learning Environment – Tutorial. No. CI-136/02, Collaborative Research Center 531, University of Dortmund, Dortmund, Germany (2002)
3. Hacker, S.: MP3, the definitive guide. O'Reilly & Associates, Inc. (2000)
4. Kohonen, T.: Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics* 43, 59–69 (1982)
5. Kohonen, T.: *Self-Organizing Maps*. Springer, New York (2001)
6. Kurth, F., Clausen, M.: Full-Text Indexing of Very Large Audio Data Bases. 110th Audio Engineering Society Convention (2001)
7. Koza, J.R.: Genetic Programming. *Encyclopedia for Computer Science and Technology* (1997)
8. Liu, Z., Wang, Y., Chen, T.: Audio Feature Extraction and Analysis for Scene Segmentation and Classification. *Journal of VLSI Signal Processing* 20, 61–79 (1998)
9. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid Prototyping for Complex Data Mining Tasks. In: KDD 2006. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York (2006)
10. Mierswa, I., Morik, K.: Automatic Feature Extraction for Classifying Audio Data. *Machine Learning Journal* 58, 127–149 (2005)
11. Mierswa, I.: Value Series Processing with Yale. Version 3.3
12. Pachet, F., Laïgre, D.: A Naturalist Approach to Music File Name Analysis. In: Proceedings of 2nd International Symposium on Music Information Retrieval (2001)
13. Pardo, B., Shlfrin, J., Birmingham, W.: Name That Tune: A Pilot Study in Finding a Melody From a Sung Query. *Journal of the American Society for Information Science and Technology* 55(4) (2004)
14. Schedl, M., Pampalk, E., Widmer, G.: Intelligent Structuring and Exploration of Digital Music Collections. Austrian Research Institute for Artificial Intelligence (ÖFAI), Vienna, Austria and Department of Computational Perception Johannes Kepler Universität (JKU) Linz, Austria (2004)
15. Tzanetakis, G., Essl, G., Cook, P.: Automatic Musical Genre Classification Of Audio Signals. In: ISMIR. Proceedings of the Int. Symposium on Music Information Retrieval, pp. 205–210 (2001)

16. Uitdenbogerd, A.L., Zobel, J.: An Architecture for Effective Music Information Retrieval. *Journal of the American Society for Information Science and Technology* 55(12), 1053–1057 (2004)
17. Unal, E., Narayanan, S.S., Chew, E.: A Statistical Approach to Retrieval under User-dependent Uncertainty in Query-by-Humming Systems. In: *International Multimedia Conference, Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 113–118. ACM Press, New York (2004)

## Appendix: Similarity Cluster Examples

In the following, please find a number of tables with a choice of similarity clusters that have evolved from the clustering process. The artist names are printed in regular font, the song titles are bold. The value in parentheses denotes the similarity to the target song ranked on a 6-point scale. On the left hand side of the table, all songs in the cluster are shown. On the right hand side, all similar songs (at least a ranking of 3 or above) that have been “forgotten” by the framework are displayed. For reference, the tables also include the basic settings of the individual test run and an enumeration of the files not included in the similarity cluster.

**Table 1.** Similarity Cluster for “Nothing else Matters” by Metallica

<i>Similar Songs in Cluster (ranking):</i>	<i>Similar songs not in cluster (ranking):</i>
Band Ohne Namen <b>Take my Heart</b> (5)	Highland <b>Bella Stella</b> (5)
Die 3. Generation <b>Ich will, dass Du mic liebst</b> (5)	Echt <b>Weinst Du</b> (5)
Laura <b>Immer wieder</b> (5)	Reamonn <b>Supergirl</b> (4)
Die Ärzte <b>Wie Es Geht</b> (4)	Orange Blue <b>She's Got That Light</b> (4)
Sisqo <b>Thong Song</b> (4)	R Kelly <b>If I could Turn back Hands</b> (4)
Madonna <b>Music</b> (3)	Rednex <b>Spirit of The Hawk</b> (3)
Anastasia <b>I'm Outa Love</b> (3)	Santana <b>Maria Maria</b> (3)
Manu Chao <b>Bongo Bong</b> (3)	Madonna <b>American Pie</b> (3)
Music Instructor feat. Dean <b>Super Fly</b> (3)	Sting <b>Desert Rose</b> (3)
Gigi D'Agostino <b>The Riddle</b> (0)	Gabrielle <b>Rise</b> (3)
Limp Bizkit <b>Take a Look Around</b> (0)	Corpus Size : 100
ATC <b>My Heart Beats like a Drum</b> (0)	Neurons : 15x15
Mauro Picotto <b>Komodo</b> (0)	Filtering : No
Vanessa Amorosi <b>Absolutely Everybody</b> (0)	Normalization : No
Alex <b>Ich will nur Dich</b> (0)	Total # of Clusters : 7
	Similar files : 15
	Target Song: Metallica <b>Nothing else Matters</b>

**Table 2.** Similarity Cluster for “Nothing else Matters” by Metallica

<i>SIMILAR SONGS IN CLUSTER (RANKING):</i>	<i>Similar songs not in cluster (ranking):</i>
Elton John & George Michael <b>Don't Let The Sun Go Down on Me</b> (5)	Band Ohne Namen <b>Take my Heart</b> (5)
Garland Jeffreys <b>Hail Hail Rock 'n Roll</b> (5)	Highland <b>Bella Stella</b> (5)
Michael Jackson <b>Heal the World</b> (5)	Echt <b>Weinst Du</b> (5)
Youssou Ndour & Neneh Cherry <b>7 Seconds</b> (5)	Die Ärzte <b>Wie Es Geht</b> (5)
Meat Loaf <b>I'd Do Anything for Love</b> (5)	Die 3. Generation <b>Ich will, dass Du mich liebst</b> (5)
Young Deenay <b>Walk On By</b> (5)	Laura <b>immer wieder</b> (5)
Thomas D <b>Liebesbrief</b> (5)	Reamonn <b>Supergirl</b> (4)
Christina Aguilera <b>Beautiful</b> (5)	Orange Blue <b>She's Got That Light</b> (4)
Coldplay <b>Speed Of Sound</b> (5)	R Kelly <b>If I could Turn back Hands</b> (4)
Hypertraxx <b>The Darkside</b> (4)	Sisqo <b>Thong Song</b> (3)
Nelly Furtado <b>I'm Like a Bird</b> (4)	Rednex <b>Spirit of The Hawk</b> (3)
Atomic Kitten <b>It's OK</b> (4)	Santana <b>Maria Maria</b> (3)
Daniel Bedingfield <b>If You're Not The One</b> (4)	Madonna <b>Music</b> (3)
Nomad <b>I wanna give you devotion</b> (4)	Madonna <b>American Pie</b> (3)
Salt 'n Pepa <b>Lets Talk about Sex</b> (3)	Sting <b>Desert Rose</b> (3)
Ace of Base <b>Don't Turn Around</b> (3)	Gabrielle <b>Rise</b> (3)
Dune <b>Hardcore Vibes</b> (3)	Anastasia <b>I'm Outa Love</b> (3)
Chris Brown <b>Run It</b> (3)	Manu Chao <b>Bongo Bong</b> (3)
Dru Hill <b>How Deep is Your Love</b> (2)	Music Instructor feat. Dean <b>Super Fly</b> (3)
J-Kwon <b>Tipsy</b> (2)	
Color Me Badd <b>I Wanna Sex You Up</b> (1)	
Cher <b>Believe</b> (1)	<i>(results truncated)</i>
Interactive <b>Living Without Your Love</b> (0)	CORPUS SIZE : 1,554
Wolfgang Petry <b>Die Längste Single</b> (0)	NEURONS : 15X15
	Filtering : No
	Normalization : No
	Total # of Clusters : 143
	Similar files : 24
	Target Song:
	Metallica <b>Nothing else Matters</b>

# Complete Recall on Alpha-Beta Heteroassociative Memory

Israel Román-Godínez and Cornelio Yáñez-Márquez

Centro de Investigación en Computación  
Juan de Dios Bátiz s/n esq. Miguel Othón de Mendizábal  
Unidad Profesional Adolfo López Mateos  
Del. Gustavo A. Madero, México, D.F. México  
`iromanb05@sagitario.cic.ipn.mx`,  
`cyanez@cic.ipn.mx`

**Abstract.** Most heteroassociative memories models intend to achieve the recall of the entire trained pattern. The Alpha-Beta associative memories only ensure the correct recall of the trained patterns in autoassociative memories, but not for the heteroassociative memories. In this work we present a new algorithm based on the Alpha-Beta Heteroassociative memories that allows, besides correct recall of some altered patterns, perfect recall of all the trained patterns, without ambiguity. The theoretical support and some experimental results are presented.

## 1 Introduction

Associative memories have been an active area for research in computer sciences. In this respect, computer scientists are interested in developing mathematical models that behave as similar as possible to associative memories and, based on the former models, create, design and operate systems that are able to learn and recall patterns [1-3]. The ultimate goal of an associative memory is to correctly recall complete patterns from input patterns. These patterns might be an altered version of the one used to create the associative memory. The first known mathematical model of an associative memory is Steinbuch's Lernmatrix, developed in 1961 [4]. In the following years, many efforts were made. By 1982, Hopfield created a model that works, simultaneously, as associative memory and a neural network [5]. In the late 1990s morphological associative memories were developed by Ritter et al. [6]. In 2002, a more efficient model of associative memories arose; the Alpha-Beta associative memories were inspired on morphological associative memories [1]. Until this day, the Alpha-Beta model has been applied to several noteworthy problems, such as automatic color matching [7] and language translators [8].

In this paper we propose an improvement on the Alfa-Beta associative memories, particularly on the heteroassociative memory, to ensure the correct recall of the fundamental set, characteristic that does not have the original model. The mathematical support is presented.

This paper is organized as follows. Sections 2 is focused on explaining the Alpha-Beta heteroassociative memory model. Section 3 contains the core proposal and its theoretical support. Section 4 is devoted to the experimental results and finally the Section 5 is about conclusions and future research.

## 2 Alpha-Beta Associative Memories

Here we use basic concepts about associative memories presented in [1]. An associative memory  $\mathbf{M}$  is a system that relates input patterns, and outputs patterns, as follows:  $\mathbf{x} \longrightarrow \boxed{\mathbf{M}} \longrightarrow \mathbf{y}$ . Each input vector  $\mathbf{x}$  forms an association with a corresponding output vector  $\mathbf{y}$ . The  $k$ -th association will be denoted as  $(\mathbf{x}^k, \mathbf{y}^k)$ . Associative memory  $\mathbf{M}$  is represented by a matrix whose  $ij$ -th component is  $m_{ij}$ , and is generated from an *a priori* finite set of known associations, called the fundamental set of associations. If  $\mu$  is an index, the fundamental set is represented as:  $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$  with  $p$  the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that  $\mathbf{x}^\mu = \mathbf{y}^\mu \forall \mu \in \{1, 2, \dots, p\}$ , then  $\mathbf{M}$  is *autoassociative*, otherwise it is *heteroassociative*. In this latter case it is possible to establish that  $\exists \mu \in \{1, 2, \dots, p\}$  for which  $\mathbf{x}^\mu \neq \mathbf{y}^\mu$ . If when feeding a unknown fundamental pattern  $\mathbf{x}^\omega$  with  $\omega \in \{1, 2, \dots, p\}$  to an associative memory  $\mathbf{M}$ , it happens that the output corresponds exactly to the associated pattern  $\mathbf{y}^\omega$ , we say that recall is correct.

The heart of the mathematical tools used in the Alpha-Beta model, are two binary operators designed specifically for these memories. These operators are defined in [1] as follows: First, we define the sets  $A = \{0, 1\}$  and  $B = \{0, 1, 2\}$ , then the operators  $\alpha : A \times A \rightarrow B$  and  $\beta : A \times B \rightarrow A$  are defined in tabular form:

**Table 1.** Alpha and Beta Operators

x	y	$\alpha(x, y)$	x	y	$\beta(x, y)$
0	0	1	0	0	0
0	1	0	0	1	0
1	0	2	1	0	0
1	1	1	1	1	1
				2	1
				2	1

Two types of heteroassociative Alpha-Beta memories are proposed: type Max ( $\vee$ ) and type Min ( $\wedge$ ). For the generation of both types we will use the operator  $\boxtimes$ , which has the following form, for indices  $\mu \in \{1, 2, \dots, p\}$ ,  $i \in \{1, 2, \dots, m\}$ , and  $j \in \{1, 2, \dots, n\}$ :  $\left[ \mathbf{y}^\mu \boxtimes (\mathbf{x}^\mu)^t \right]_{ij} = \alpha(y_i^\mu, x_j^\mu)$ .

### Alpha-Beta Heteroassociative Memories Type Max

*Learning Phase.* For every  $\mu = 1, 2, \dots, p$ , from the pair  $(\mathbf{x}^\mu, \mathbf{y}^\mu)$  build the matrix:  $\left[ \mathbf{y}^\mu \boxtimes (\mathbf{x}^\mu)^t \right]_{m \times n}$ . Applying the Max binary operator  $\vee$ , the  $\mathbf{V}$  matrix

is:  $\mathbf{V} = \bigvee_{\mu=1}^p [\mathbf{y}^\mu \boxtimes (\mathbf{x}^\mu)^t]$  then the  $ij$ -th entry is given as:  $\nu_{ij} = \bigvee_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu)$ . We can observe that  $\nu_{ij} \in B, \forall i \in \{1, 2, \dots, m\}, \forall j \in \{1, 2, \dots, n\}$ .

*Recall Phase.* A pattern  $\mathbf{x}^\omega$ , that could be or not from the fundamental set, is presented to the heteroassociative Alpha-Beta memory of type  $\vee$  and we do the operation  $\Delta_\beta: \mathbf{V} \Delta_\beta \mathbf{x}^\omega$ . The result is a column vector of dimension  $m$ , whose  $i$ -th component is:  $(\mathbf{V} \Delta_\beta \mathbf{x}^\omega)_i = \bigwedge_{j=1}^n \beta(\nu_{ij}, x_j^\omega)$ .

*Remark 1.* The Alpha-Beta Heteroassociative Memory Type Min are developed by duality, based on the learning and recall phase of the Alpha-Beta heteroassociative memories type Max. Wherever there is a  $\vee$  operator change it for  $\bigwedge$ , if there is an  $\bigwedge$  change it for  $\bigvee$ , where the operator  $\Delta_\beta$  is used change it for  $\nabla_\beta$ .

### 3 Alpha-Beta Heteroassociative Memories with Complete Recall

The Alpha-Beta autoassociative memories guarantee the complete recall of the fundamental set [20], but in the case of the heteroassociative memories is not possible to ensure this behavior. In this section we propose a new algorithm, modifying the original, with which the complete recall of the fundamental set is guaranteed.

**Definition 1.** Let  $h, n \in \mathbb{Z}^+, A = \{0, 1\}$  and let  $\mathbf{x}^h \in A^n$  be a binary pattern.

We denote the sum of the positive components of  $\mathbf{x}^h$  by:  $U_h = \sum_{j=1}^n x_j^h$ .

**Definition 2.** Let  $\mathbf{V}$  be an Alpha-Beta heteroassociative memory type Max and  $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$  its fundamental set with  $\mathbf{x}^\mu \in A^n$  and  $\mathbf{y}^\mu \in A^p, A = \{0, 1\}, B = \{0, 1, 2\}, n \in \mathbb{Z}^+$ . The sum of the components with value equal to one of the  $i$ -th row of  $\mathbf{V}$  is given as:  $s_i = \sum_{j=1}^n T_j$  where  $T \in B^n$  and its components

are defined as:  $T_i = \begin{cases} 1 & \iff \nu_{ij} = 1 \\ 0 & \iff \nu_{ij} \neq 1 \end{cases} \forall j \in \{1, 2, \dots, n\}$  and the  $s_i$  components conform the max sum vector with  $\mathbf{s} \in \mathbb{Z}^p$ .

**Definition 3.** Let  $\alpha, \beta, n \in \mathbb{Z}^+, A = \{0, 1\}$  and let  $\mathbf{x}^\alpha, \mathbf{x}^\beta \in A^n$  be two vectors; then  $\mathbf{x}^\alpha \leq \mathbf{x}^\beta \iff x_i^\alpha = 1 \rightarrow x_i^\beta = 1 \forall i \in \{1, 2, \dots, n\}$  and  $\mathbf{x}^\alpha < \mathbf{x}^\beta \iff \forall i x_i^\alpha \leq x_i^\beta$  and  $\exists j$  such that  $x_j^\alpha < x_j^\beta$

**Definition 4.** Let  $\mathbf{x}^\alpha \in A^n$  with  $\alpha, n \in \mathbb{Z}^+, A = \{0, 1\}$ ; each component of the negated vector, denoted by  $\sim \mathbf{x}^\alpha$ , of  $\mathbf{x}^\alpha$  is given as:  $\sim x_i^\alpha = \begin{cases} 1 & x_i^\alpha = 0 \\ 0 & x_i^\alpha = 1 \end{cases} \forall i \in \{1, 2, \dots, n\}$ .

**Definition 5.** Let  $\alpha, \beta, n \in \mathbb{Z}^+, A = \{0, 1\}$  and let be  $\mathbf{x}^h \in A^n$ . We denote the sum of the components equal to 0 of  $\mathbf{x}^h$  as:  $C_h = \sum_{i=1}^n \sim x_i^h$ .

**Definition 6.** Let  $\Lambda$  be a Alpha-Beta heteroassociative memory type Min and  $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$  its fundamental set with  $\mathbf{x}^\mu \in A^n$  and  $\mathbf{y}^\mu \in A^p$ ,  $A = \{0, 1\}$ ,  $B = \{0, 1, 2\}$ ,  $n \in Z^+$ . The sum of the components with value equal to zero of the  $i$ -th row of  $\Lambda$  is given as:  $r_i = \sum_{j=1}^n T_j$  where  $T \in B^n$  and its components are defined as:  $T_i = \begin{cases} 1 & \longleftrightarrow \lambda_{ij} = 0 \\ 0 & \longleftrightarrow \lambda_{ij} \neq 0 \end{cases} \forall j \in \{1, 2, \dots, n\}$  and the  $r_i$  components conform the min sum vector with  $\mathbf{r} \in Z^p$ .

### Alpha-Beta Heteroassociative Memory Type Max

*Learning Phase.* Let  $\mathbf{x} \in A^n$  and  $\mathbf{y} \in A^p$  be an input and output vectors, respectively. The corresponding fundamental set is denoted by  $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$  which is built according with the following conditions: the  $\mathbf{y}$  vectors are built with the *one-hot* codification: assigning for  $\mathbf{y}^\mu$  the following values:  $y_k^\mu = 1$ , and  $y_j^\mu = 0$  for  $j = 1, 2, \dots, k - 1, k + 1, \dots, m$  where  $k \in \{1, 2, \dots, m\}$ . And to each  $\mathbf{y}^\mu$  vector correspond *one and only one*  $\mathbf{x}^\mu$  vector.

*Step 1.* For each  $\mu \in \{1, 2, \dots, p\}$ , from the couple  $(\mathbf{x}^\mu, \mathbf{y}^\mu)$  build the matrix:  $[\mathbf{y}^\mu \boxtimes (\mathbf{x}^\mu)^t]_{m \times n}$  then, the min binary operator  $\vee$  is applied to the matrices.

Therefore, the  $\mathbf{V}$  matrix is obtained as follow:  $\mathbf{V} = \bigvee_{\mu=1}^p [\mathbf{y}^\mu \boxtimes (\mathbf{x}^\mu)^t]$  where the

$ij$ -th component is given by:  $v_{ij} = \bigvee_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu)$ .

### Recalling Phase

*Step 1.* A pattern  $\mathbf{x}^\varpi$  is presented to  $\mathbf{V}$ , the  $\Delta_\beta$  operation is done and the resulting vector is assigned to a vector called  $\mathbf{z}^\varpi$ :  $\mathbf{z}^\varpi = \mathbf{V} \Delta_\beta \mathbf{x}^\varpi$ . Then the  $i$ -th component of  $\mathbf{z}^\varpi$  is:  $z_i^\varpi = \bigwedge_{j=1}^n \beta(v_{ij}, x_j^\varpi)$

*Step 2.* Once we have the  $\mathbf{V}$  matrix, it is necessary to build the *max sum vector*  $\mathbf{s}$  according to the definition 2, therefore the corresponding  $\mathbf{y}^\varpi$  is given as:

$$\mathbf{y}_i^\varpi = \begin{cases} 1 & \text{if } s_i = \bigvee_{k \in \theta} s_k \wedge z_i^\varpi = 1 \\ 0 & \text{otherwise} \end{cases}$$

where  $\theta = \{i \mid z_i^\varpi = 1\}$ .

Below are presented the lemmas, and a theorem that support the Alpha-Beta heteroassociative memory type Max presented before.

**Lemma 1.** Let  $x^i \in A^n$  be a pattern randomly chosen from the fundamental set. In the new Alpha-Beta heteroassociative memory type Max learning phase,  $x^i$  contributes, only, at the  $i$ -th row of  $\mathbf{V}$  with  $U_i$  times the value 1 and  $(n - U_i)$  times the value 0.



*Proof.* Let  $x^h \in A^n$  and  $y^h \in A^p$  with  $A = \{0, 1\}$  and  $k, n, p \in Z^+$  be two fundamental patterns, randomly chosen, that form the  $k$ -th association  $(x^k, y^k)$  of  $\mathbf{V}$ . According with the learning phase we know that the matrix  $\mathbf{V}$  is given

by:  $\mathbf{V} = \bigvee_{\mu=1}^p [\mathbf{y}^\mu \boxtimes (\mathbf{x}^\mu)^t]$  particularly the  $k$ -th association is:  $[\mathbf{y}^k \boxtimes (\mathbf{x}^k)^t]_{ij} = \alpha(\mathbf{y}_i^k, \mathbf{x}_j^k)$ . Now, by how the vector  $\mathbf{y}^k$  has been built, it happend that  $\forall i \in \{1, 2, \dots, k-1, k+1, \dots, p\}, \forall j \in \{1, 2, \dots, n\}, k \in \{1, 2, \dots, p\}$

$$\begin{aligned} \mathbf{y}_k^k &= 1 \rightarrow \alpha(\mathbf{y}_k^k, \mathbf{x}_j^k) = 1 \vee \alpha(\mathbf{y}_k^k, \mathbf{x}_j^k) = 2 \\ \mathbf{y}_i^k &= 0 \rightarrow \alpha(\mathbf{y}_i^k, \mathbf{x}_j^k) = 1 \vee \alpha(\mathbf{y}_i^k, \mathbf{x}_j^k) = 0 \end{aligned} \tag{1}$$

according to expression [1](#) it is evident that the maximum values of the  $k$ -th matrix are stored in its  $k$ -th row, depending exclusively on the values of  $\mathbf{x}^k$ , in other words, when  $\mathbf{x}_j^k = 1 \iff \alpha(\mathbf{y}_k^k, \mathbf{x}_j^k) = 1$  or  $\mathbf{x}_j^k = 0 \iff \alpha(\mathbf{y}_k^k, \mathbf{x}_j^k) = 2$ . Therefore, considering that for every fundamental association, *to each input pattern correspond one and only one output pattern* and that  $k$  with  $k \in \{1, 2, \dots, p\}$  was randomly chosen; we can ensure that  $\mathbf{V}$  is affected in its  $i$ -th row by  $\mathbf{x}^i$  and it is affected with  $U_i$  times the value 1 and  $(n - U_i)$  times the value 2. Thus, the components of the  $\mathbf{V}$  matrix contain only the values 1 or 2. Finally, we can rewrite the learning phase as follow:  $\forall i \in \{1, 2, \dots, p\}, \forall j \in \{1, 2, \dots, n\}$

$$v_{ij} = \alpha(\mathbf{y}_i^i, \mathbf{x}_j^i) \tag{2}$$

■

**Lemma 2.** *Let  $\mathbf{s}$  be the max sum vector of the matrix  $\mathbf{V}$ ; then  $s_i = U_i \forall i \in \{1, 2, \dots, p\}$*

*Proof.* Let  $\mathbf{s}$  be the *max sum vector* of the matrix  $\mathbf{V}$ . Its  $i$ -th component is expressed as definition [2](#)

$$s_i = \sum_{j=1}^n T_j \tag{3}$$

In the other hand, we know by definition [1](#) that  $U_h = \sum_{j=1}^n x_j^h$ . Particularly, for a  $i$  with  $i \in \{1, 2, \dots, p\}$  the expression could be written as follow:

$$U_i = \sum_{j=1}^n x_j^i \tag{4}$$

Moreover, we know by lemma [1](#) in expression [2](#) that  $x^i$  affects the matrix  $\mathbf{V}$  only in its  $i$ -th row, so it is possible to rewrite the expression [3](#) as:  $s_i = \sum_{j=1}^n \alpha(y_i^i, x_j^i)$ .

Given that  $y_i^i = 1 \forall i \in \{1, 2, \dots, p\}$  and that  $\alpha(y_i^i, x_j^i)$  depends on  $x_j^i$ , then according to lemma [1](#)

$$s_i = \sum_{j=1}^n x_j^i \tag{5}$$

Finally, by transitivity of the equations [4](#) and [5](#) we can conclude that  $s_i = \sum_{j=1}^n x_j^i = U_i$ . ■

**Lemma 3.** *Let  $\mathbf{V}$  be a heteroassociative memory type Max which fundamental set is  $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$ , then let  $\mathbf{x}^\varpi \in A^n$  be pattern that will be presented to  $\mathbf{V}$  with  $A \in \{0, 1\}$ ,  $\varpi \in \{1, 2, \dots, p\}$ ,  $n, p \in \mathbb{Z}^+$ . The  $z^\varpi \in A^p$  vector obtained from the original Alpha-Beta heteroassociative recall phase type Max will contain the value 1 in its  $i$ -th component where the  $i$ -th row of  $\mathbf{V}$  correspond to the fundamental patterns lower or equal to  $\mathbf{x}^\varpi$ ; put differently:  $\forall i, z_i^\varpi = 1 \rightarrow x^i \leq x^\omega, x^i \in \{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$ .*

*Proof.* According with the original Alpha-Beta heteroassociative memory recall phase type Max we know that:

$$z_i^\varpi = 1 \longrightarrow \bigwedge_{j=1}^n \beta(v_{ij}, x_j^\varpi) = 1 \tag{6}$$

in order to  $\bigwedge_{j=1}^n \beta(v_{ij}, x_j^\varpi) = 1$ , due to  $\beta$  only produce 1 or 2 values, it is necessary that  $\forall j \in \{1, 2, \dots, n\} \beta(v_{ij}, x_j^\varpi) = 1$ . Therefore, considering lemma [1](#),  $\forall j \in \{1, 2, \dots, n\}$  just the following cases are possible

$$\beta(v_{ij}, x_j^\varpi) = 1 \longrightarrow \begin{cases} v_{ij} = 1 \wedge x_j^\varpi = 1 \\ v_{ij} = 2 \wedge (x_j^\varpi = 1 \vee x_j^\varpi = 0) \end{cases} \tag{7}$$

Now, as lemma [1](#) says, each  $\mathbf{x}^i$  pattern affect only the  $i$ -th row of  $\mathbf{V}$  and it does according to learning phase, from the expression [7](#) we can infer that:  $x^i = x^\varpi$  if  $\forall j$  always happend that  $(v_{ij} = 1 \wedge x_j^\varpi = 1)$  or  $(v_{ij} = 2 \wedge x_j^\varpi = 0)$  and  $x^i < x^\varpi$  if  $\forall j$  always happend that  $(v_{ij} = 1 \wedge x_j^\varpi = 1)$  or  $\exists j(v_{ij} = 2 \wedge x_j^\varpi = 1)$ . This is

$$x^i \leq x^\varpi \tag{8}$$

therefore, by transitivity of [6](#), [7](#), [8](#) we can conclude:

$$\forall i, z_i^\varpi = 1 \rightarrow x^i \leq x^\varpi, x^i \in \{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\} \tag{9}$$

**Theorem 1.** *Let  $\mathbf{V}$  be a heteroassociative memory type Max which fundamental set is  $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$ , without any pair repeted. Let  $\mathbf{x}^\varpi \in A^n$  be an input pattern presented to  $\mathbf{V}$  and  $\mathbf{z}^\varpi \in A^p$  the resulting class vector from the original Alpha-Beta heteroassociative memory recall phase type Max. The proposed algorithm will always obtain complete recall, in other word, we will always obtain the corresponding  $\mathbf{y}^\varpi$  without ambiguity.*

*Proof.* To prove the complete recall of the proposed algorithm it would be necessary to ensure that, for all components where  $z_i^\varpi = 1$  there is just one maximum value in the  $s_i$  components and it correspond to the correct pattern. This could

be demonstrated by contradiction. Let  $x^\alpha \in A^n$  and  $x^\beta \in A^n$  be the corresponding patterns to  $z_\alpha^\varpi = 1$  and  $z_\beta^\varpi = 1$  when  $x^\varpi$  is presented to  $\mathbf{V}$  with  $x^\alpha, x^\beta, x^\varpi \in \{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$ . First, we assume that  $x^\alpha$  is the correct pattern and  $x^\beta$  is an arbitrary spurious recalled pattern with corresponding  $s_i$  values  $s_\alpha$  and  $s_\beta$ , respectively and it holds that  $s_\alpha > s_\beta$ . Now, we assume the negated of what we want to prove.

$$s_\alpha \leq s_\beta \tag{9}$$

Now, by lemma 2, we know that the expression 9 could be written as follow:

$$U_\alpha \leq U_\beta \tag{10}$$

By lemma 3 for each spurious pattern  $x^i$ , where  $x^\varpi$  is the correct, imply that  $\forall i z_i^\varpi = 1, \mathbf{x}^i \neq \mathbf{x}^\varpi \rightarrow \mathbf{x}^i < \mathbf{x}^\varpi$ . Therefore, we can take as a hypothesis:

$$\mathbf{x}^\beta < \mathbf{x}^\alpha \tag{11}$$

according to definition 1, the inequality 11 could be expressed as  $U_\beta < U_\alpha$  which is a contradiction with expression 10, then  $U_\alpha \leq U_\beta$  is false. Therefore,  $U_\alpha > U_\beta$ , put differently  $s_\alpha > s_\beta$ , is true for every spurious recalled pattern since  $x^\beta$  was chosen arbitrarily. ■

### Alpha-Beta Heteroassociative Memory Type Min

*Learning Phase.* Let  $\mathbf{x} \in A^n$  and  $\mathbf{y} \in A^p$  be input and output vectors, respectively. The corresponding fundamental set is denoted by  $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$ . which is built according with the following conditions: the  $\mathbf{y}$  vectors are built with the *zero-hot* codification: assigning for the output binary pattern  $\mathbf{y}^\mu$  the following values:  $y_k^\mu = 0$ , and  $y_j^\mu = 1$  for  $j = 1, 2, \dots, k - 1, k + 1, \dots, m$  where  $k \in \{1, 2, \dots, m\}$ . And, to each  $\mathbf{y}^\mu$  vector correspond *one and only one*  $\mathbf{x}^\mu$  vector.

*Step 1.* For each  $\mu \in \{1, 2, \dots, p\}$ , from the couple  $(\mathbf{x}^\mu, \mathbf{y}^\mu)$  build the matrix:  $[\mathbf{y}^\mu \boxtimes (\mathbf{x}^\mu)^t]_{m \times n}$  then the Min binary operator ( $\wedge$ ) is applied to the matrices obtained. Therefore, the  $\mathbf{\Lambda}$  matrix is obtained as follow:  $\mathbf{\Lambda} = \bigwedge_{\mu=1}^p [\mathbf{y}^\mu \boxtimes (\mathbf{x}^\mu)^t]$

where the  $ij$ -th component is given by:  $\lambda_{ij} = \bigwedge_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu)$ .

#### Recalling Phase

*Step 1.* A pattern  $\mathbf{x}^\varpi$  is presented to  $\mathbf{\Lambda}$ , the  $\nabla_\beta$  operation is done and the resulting vector is assigned to a vector called  $\mathbf{z}^\varpi$ :  $\mathbf{z}^\varpi = \mathbf{\Lambda} \nabla_\beta \mathbf{x}^\varpi$  The  $i$ -th component of the resulting column vector are:  $\mathbf{z}_i^\varpi = \bigwedge_{j=1}^n \beta(\lambda_{ij}, x_j^\varpi)$

*Step 2.* It is necessary to build the *min sum vector*  $\mathbf{r}$  according to the definition **6**, therefore the corresponding  $\mathbf{y}^\varpi$  is given as:

$$\mathbf{y}_i^\varpi = \begin{cases} 0 & \text{if } r_i = \bigwedge_{k \in \theta} r_k \wedge z_i^\varpi = 0 \\ 1 & \text{otherwise} \end{cases}$$

where  $\theta = \{i | z_i^\varpi = 0\}$ .

Below are presented the lemmas, and a theorem that support the Alpha-Beta heteroassociative memory type Min presented before. Due to a matter of space, the proof of the lemma **4**, **5** and **6** are not developed here, but they were obtained by duality.

**Lemma 4.** *Let  $x^i \in A^n$  be a pattern randomly chosen from the fundamental set. In the Alpha-Beta heteroassociative memory type Min learning phase,  $x^i$  contributes, only, at the  $i$ -th row of  $\mathbf{\Lambda}$  with  $U_i$  times the value 0 and  $(n - U_i)$  times the value 1.*

*Proof.* This proof is similar to the one presented on lemma **1** taking account the conditions expressed on remark 1. ■

**Lemma 5.** *Let  $\mathbf{r}$  be the min sum vector of the matrix  $\mathbf{\Lambda}$ ; then  $r_i = C_i, \forall i \in \{1, 2, \dots, p\}$ .*

*Proof.* This proof is similar to the one presented on lemma **2** taking account the conditions presented on remark 1. ■

**Lemma 6.** *Let  $\mathbf{\Lambda}$  be a heteroassociative memory type Min which fundamental set is  $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$ . Let  $\mathbf{x}^\varpi \in A^n$  be an input pattern that will be presented to  $\mathbf{\Lambda}$  with  $A \in \{0, 1\}, \varpi \in \{1, 2, \dots, p\}, n, p \in \mathbb{Z}^+$ . The  $z^\varpi \in A^p$  vector is obtained from the original Alpha-Beta heteroassociative recall phase type Min. This vector will contain the value 0 in its  $i$ -th component where the  $i$ -th row of  $\mathbf{\Lambda}$  correspond to the fundamental patterns greater or equal to  $\mathbf{x}^\varpi$ ; put differently:  $\forall i \ z_i^\varpi = 0 \rightarrow x^i \geq x^\omega, x^i \in \{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$ .*

*Proof.* This proof is similar to the one presented on lemma **3** taking account the conditions expressed on remark 1. ■

**Theorem 2.** *Let  $\mathbf{\Lambda}$  be a heteroassociative memory type Min which fundamental set, without any pair repeated, is  $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$ . Let  $\mathbf{x}^\varpi \in A^n$  be an input pattern presented to  $\mathbf{\Lambda}$  and  $\mathbf{z}^\varpi \in A^p$  the resulting class vector from the original Alpha-Beta heteroassociative memory recall phase type Min. The proposed algorithm will always obtain complete recall, in other word, we will always obtain the corresponding  $\mathbf{y}^\varpi$  without ambiguity.*

*Proof.* To prove the complete recall of the proposed algorithm it would be necessary to ensure that, for all components where  $z_i^\varpi = 0$  there is just one maximum value in the  $r_i$  components and it correspond to the correct pattern. This could

be demonstrated by contradiction. Let  $x^\alpha \in A^n$  and  $x^\beta \in A^n$  be the corresponding patterns to  $z_\alpha^\varpi = 0$  and  $z_\beta^\varpi = 0$  when  $x^\varpi$  is presented to  $\mathbf{\Lambda}$  with  $x^\alpha, x^\beta, x^\varpi \in \{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, p\}$ . First, we assume that  $x^\alpha$  is the correct pattern and  $x^\beta$  is an arbitrary spurious recalled pattern with corresponding  $r_i$  values  $r_\alpha$  and  $r_\beta$ , respectively and it holds that  $r_\alpha < r_\beta$ . Now, we assume the negated of what we want to prove.

$$s_\alpha \geq s_\beta \tag{12}$$

Now, by lemma 5, we know that the expression 12 could be written as follow:

$$C_\alpha \geq C_\beta \tag{13}$$

By lemma 6 for each spurious pattern  $x^i$ , where  $x^\varpi$  is the correct, imply that  $\forall i z_i^\varpi = 0, \mathbf{x}^i \neq \mathbf{x}^\varpi \rightarrow \mathbf{x}^i > \mathbf{x}^\varpi$ . Therefore, we can consider as a hypothesis:

$$\mathbf{x}^\beta > \mathbf{x}^\alpha \tag{14}$$

according to definition 5, the inequality 14 could be expressed as  $U_\beta > U_\alpha$  which is a contradiction with expression 13, then  $C_\alpha \geq C_\beta$  is false. Therefore,  $C_\alpha < C_\beta$ , put differently  $r_\alpha > r_\beta$ , is true for every spurious recalled pattern since  $x^\beta$  was chosen arbitrarily. ■

## 4 Experimental Results

In despite of the theoretical support presented in the last section, a serie of experiments were done to illustrate the efficiency of our proposal. With  $n$  the dimension of the input vector and  $p$  the number of input patterns, three different finite samples were randomly generated, automatically. Each of them was used to build the six different fundamental sets according with the specification mentioned in our proposal. After that, the six different memories -three Max and three Min- were built and the recall phase, each one with its corresponding fundamental patterns, was applied. The results are presented in table 2.

It is evident that the original algorithm presents more errors than the one proposed in this paper.

**Table 2.** Experimental Results

Experiment Number	$n$	$p$	Original Algorithm		Modified Algorithm	
			Error (%)		Error (%)	
			Max	Min	Max	Min
1	11	100	73	76	0	0
2	15	200	63	65.5	0	0
3	6	25	76	84	0	0

## 5 Conclusion and Future Work

In this work we proposed a new algorithm for the Alpha-Beta heteroassociative memories that let us recall the fundamental patterns without ambiguity. Therefore, the model of Alpha-Beta associative memories ensure the complete recall for the fundamental set in both cases. However, the conditions for this correct recall on non-fundamental patterns has not yet characterized. The theoretical support for this proposal is presented here along with some experimental test. Currently we are working on applications of the new Alpha-Beta heteroassociative algorithm and as a future work, we will investigate which are the condition that allow our algorithm to show correct recall on non-fundamental patterns.

**Acknowledgements.** The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, and CIC), the CONACyT, and SNI for their economical support to develop this work.

## References

1. Yáñez-Márquez, C.: Associative Memories Based on Order Relations and Binary Operators (in Spanish). PhD Thesis. Center for Computing Research, México (2002)
2. Kohonen, T.: Self-Organization and Associative Memory. Springer, Heidelberg (1989)
3. Hassoun, M.H.: Associative Neural Memories. Oxford University Press, New York (1993)
4. Steinbuch, K.: Die Lernmatrix, *Kybernetik* 1(1), 26–45 (1961)
5. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. In: Proceedings of the National Academy of Sciences, vol. 79, pp. 2554–2558 (1982)
6. Ritter, G.X., Sussner, P., Diaz-de-Leon, J.L.: Morphological Associative Memories. *IEEE Transactions on Neural Networks*. 9, 281–293 (1998)
7. Yáñez-Márquez, C., Felipe-Riverón, E.M., López-Yáñez, I., Flores-Carapia, R.: A Novel Approach to Automatic Color Matching, *Lecture Notes in Computer Science*. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 302–9743. Springer, Heidelberg (2006)
8. Acevedo-Mosqueda, M.E., Yáñez-Márquez, C., López-Yáñez, I.: Alpha-Beta Bidirectional Associative Memories Based Translator. *IJCSNS International Journal of Computer Science and Network Security* 6(5A), 190–194 (2006)

# I-Cog: A Computational Framework for Integrated Cognition of Higher Cognitive Abilities


Kai-Uwe Kühnberger, Tonio Wandmacher, Angela Schwering,  
Ekaterina Ovchinnikova, Ulf Krumnack, Helmar Gust, and Peter Geibel

Institute of Cognitive Science, University of Osnabrück  
D-49076 Osnabrück, Germany

**Abstract.** There are several challenges for AI models of higher cognitive abilities like the profusion of knowledge, different forms of reasoning, the gap between neuro-inspired approaches and conceptual representations, the problem of inconsistent data, and the manifold of computational paradigms. The I-Cog architecture – proposed as a step towards a solution for these problems – consists of a reasoning device based on analogical reasoning, a rewriting mechanism operating on the knowledge base, and a neuro-symbolic interface for robust learning from noisy data. I-Cog is intended as a framework for human-level intelligence (HLI).

## 1 Introduction

Historically, artificial intelligence has (more or less) been strongly committed to interdisciplinary research and the modeling of higher cognition. Several important achievements can be identified during the last 50 years with respect to modeling (or supporting) cognitive challenging tasks of humans: state-of-the-art computer programs beat world-class chess champions or intelligent programs support our daily life in various respects, for example, when driving a car, flying a plane, or searching the web for information. Despite these apparent examples of success, there are also severe problems: there is not even an idea of how human-level intelligence (HLI) in the large can be achieved, taking into account the various forms of human capabilities, for example, concerning reasoning, problem solving, learning, adapting, acting etc. Three classes of problems are discussed in this paper: First, the problem of constant updates of knowledge, second, the variety of types of reasoning of human cognition, and third, the gap between neuro-inspired learning approaches and symbolic representations.

We think that these challenges are at the heart of achieving HLI, because of the following fundamental problem: The more fine-grained the methods are in developing tools for particular (and isolated) AI applications, the more we depart from the goal of achieving HLI and a unified model of higher cognition. 

<sup>1</sup> This claim clearly does not mean that other challenges for modeling cognition are simple or somehow straightforward to solve. Obviously open problems in computer vision or the modeling of autonomous agents and motor control, are also hard problems. But they concern lower cognitive abilities and are not aspects of HLI.

We propose an integration architecture as a model for higher cognitive abilities by the resolution of the three mentioned sub-problems.

This paper has the following structure: Section 2 sketches some problems in modeling a variety of higher cognitive abilities. Section 3 presents the I-Cog architecture consisting of an analogy engine (*AE*), an ontology rewriting device (*ORD*), and a neuro-symbolic learning device (*NSLD*). These modules interact in a non-trivial way as described in Section 4. Finally, Section 5 summarizes related work and Section 6 concludes the paper.

## 2 Problems for Modeling Higher Cognition in AI Systems

### 2.1 Knowledge

Knowledge representation is classically understood as encoding entities in the environment using symbols. Although a logical representation is generally accepted as an appropriate framework for this task, there is a non-trivial challenge for the logical approach: Whereas background knowledge is usually considered to be static, human agents constantly update, modify, and learn new conceptualizations, and they can overwrite existing knowledge easily without being threatened by inconsistency problems. Dynamic rewriting of knowledge is hard to model in AI and therefore a challenge, in order to reach a theory of HLI.

### 2.2 Reasoning

There is a manifold of human reasoning abilities. Humans perform not only deductions, inductions, and abductions, but are also able to perform analogical reasoning steps, non-monotonic inferences, and frequency-based inferences. Additionally, human agents can reason with vague and uncertain knowledge. As a natural consequence of this variety of reasoning types AI research developed a tremendous number of frameworks for the computation of inferences. Unfortunately, these computational paradigms are not fully compatible with each other.

### 2.3 Neuro-Symbolic Integration

The gap between robust neural learning and symbolic representation formalisms is obvious: Whereas symbolic theories are based on recursion and compositionality allowing the computation of (potentially) infinitely many meanings from a finite basis, such principles are not available for connectionist networks. On the other hand, neural networks have proven to be a robust tool for learning from noisy data, pattern recognition, and handling vague knowledge – classical domains with which symbolic theories usually have their problems. A potential solution for achieving HLI would require an integration of both approaches in one architecture.



### 3 Modules of the I-Cog Architecture

#### 3.1 Analogy Engine (AE)

It is a crucial hypothesis of this paper that the establishment of analogical relations between a source and a target domain can be used for many forms of classical and non-classical reasoning tasks [7]. Examples for application domains, where analogical reasoning was successfully applied, are string domains [18], geometric figures [27], problem solving [1], naive physics [5], or metaphoric expressions [13]. Furthermore analogies are a source of creativity [19] and a possibility to learn (inductively) from sparse data [12]. Vagueness plays a crucial role in every account of analogical reasoning. Deductions and abductions are implicitly modeled in several systems [6].

A rather expressive theory for computing analogies in a variety of domains is provided by heuristic-driven theory projection (HDTP) [13]. HDTP represents the source and target domains by sets of first-order formulas. The corresponding source theory  $Th_S$  and target theory  $Th_T$  are then generalized using an extension of anti-unification [26]. Here are some key elements of HDTP:

- Two formulas  $p(a)$  and  $p(b)$  can be anti-unified by  $p(X)$ , where the variable  $X$  is substituted by  $\Theta_1(X) = a$  and  $\Theta_2(X) = b$  (first-order anti-unification).
- Two formulas  $p_1(a)$  and  $p_2(a)$  can be anti-unified by  $P(a)$ , where the predicate variable  $P$  is substituted by  $\Theta_1(P) = p_1$  and  $\Theta_2(P) = p_2$  (second-order anti-unification).
- A theorem prover allows the re-representation of formulas.
- Whole theories can be generalized, not only single terms or formulas.

We exemplify analogy making in HDTP using a simple example.<sup>2</sup> Consider the following analogy induced by a metaphoric expression:

*Current is the water in the electric circuit.*

The metaphor establishes a new conceptualization of an electric circuit (target domain) by the association of *electric circuit* and *water-pipe system*, *current* and *water*, *pump* and *battery* etc. dependent on the available background knowledge. HDTP computes these associations together with a generalized theory and substitutions governed by heuristics (e.g. the complexity of relevant substitutions). Notice that by establishing a new conceptualization of the target domain, the systems learns new concepts, needs to modify the definitions and relational restrictions of hitherto concepts, or end up in an inconsistency.

The following list sketches some reasons for the major claim of this subsection, namely that a large variety of human reasoning mechanisms can be modeled by analogies.

---

<sup>2</sup> A formalization of the following analogy, as well as the specification of the underlying algorithm can be found in [12].

- Systems like HDTP allow the computation of analogical relations.
- Establishing analogical relations requires often the re-representation of a domain. HDTP achieves this by using a theorem prover included in the system, i.e. the systems allows logical deductions.
- Learning generalizations is a first step towards an induction on given input data [12]. In the example, a new conceptualization of the target is learned.
- The fact that analogies are at most psychologically preferred, but never true or false, allows the extension of the system to model uncertainty.
- Non-monotonicity can be considered as a special case of re-conceptualizing a given domain very similar to a new conceptualization induced by an analogy.

### 3.2 Ontology Rewriting Device (ORD)

The fact that human beings are able to dynamically adapt background knowledge on-the-fly was mentioned as a major challenge for HLI (cf. Section 2). We sketch some ideas (based on [23] and [24]) of a rewriting system that is constantly adapting the ontological knowledge base (memory) focusing on the resolution of inconsistencies. Although the framework was originally developed for text technological applications, the underlying logical basis is rather weak, and not all types of inconsistencies can be automatically resolved, we think that proposals in this direction are crucial for achieving HLI.

Ontological knowledge is usually formalized within a logical framework, most often in the framework of Description Logics (DL) [2]. However, the storage of ontological information within a logical framework has an undesirable side-effect: updates can cause inconsistency problems, because items of information may be contradicting, making the given ontology unsatisfiable and useless for reasoning purposes.

Ontologies usually contain a terminological and an assertion component. A DL terminology consists of a set of terminological axioms defining concepts by formulas of the form  $\forall x : C(x) \rightarrow D(x)$ , where  $C$  is a concept name and  $D$  is a concept description, i.e. a logical formula.<sup>3</sup> The assertion component mentioned above contains information about the assignment of the particular individuals to concepts and relations from the terminology. Axioms are interpreted by an interpretation function mapping concept descriptions to subsets of the domain. A model of an ontology is an interpretation satisfying all axioms. An ontology is inconsistent if it does not have a model.

There are several possibilities how inconsistencies can occur in ontologies [15]. In particular, logical inconsistencies – potentially caused by dynamic updates of the knowledge base – are of interest in our context and are addressed by an automatic rewriting device allowing constant learning and updates of the ontological knowledge base. One aspect of logical inconsistency problems concerns polysemy: If an ontology is updated automatically, then it is hardly possible to distinguish between word senses. Suppose, the concept *tree* is declared to be a subconcept both of *plant* and of *data structure* (where *plant* and *data structure*

<sup>3</sup> Compare [2] for a exhaustive definition of description logics.

are disjoint concepts). Both of these two interpretations of *tree* are correct, but it is still necessary to describe in the ontology two different concepts with different identifiers (e.g. *TreePlant*, *TreeStructure*). Another important aspect of logical inconsistency concerns generalization mistakes.

**Example 1.** Assume the following axioms are given:

$$\begin{aligned} \forall x : Bird(x) \rightarrow CanFly(x) & \quad \forall x : CanFly(x) \rightarrow CanMove(x) \\ \forall x : Canary(x) \rightarrow Bird(x) & \quad \forall x : Penguin(x) \rightarrow Bird(x) \wedge \neg CanFly(x) \end{aligned}$$

In Example 1, the statement “birds can fly” is too general. If an exception occurs (*penguin*), the ontology becomes unsatisfiable, since penguin is declared to be a bird, but it cannot fly. We will illustrate the regeneration of the overgeneralized concept *Bird* in Example 2.

**Example 2.** Adapted ontology from Example 1:

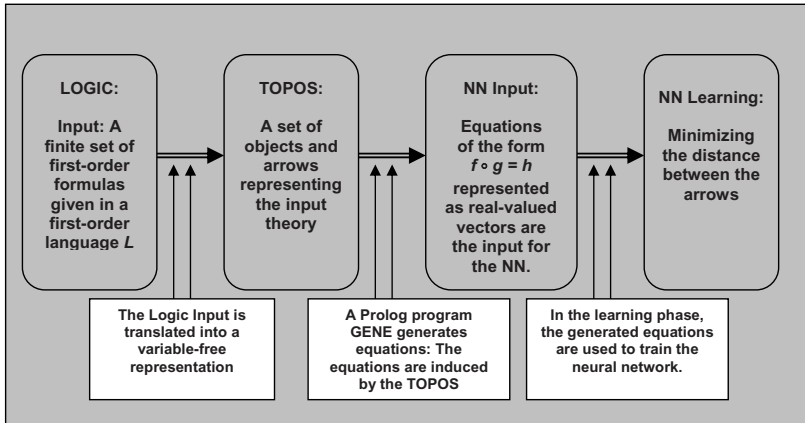
$$\begin{aligned} \forall x : Bird(x) \rightarrow CanMove(x) & \quad \forall x : CanFly(x) \rightarrow CanMove(x) \\ \forall x : Canary(x) \rightarrow FlyingBird(x) & \quad \forall x : Penguin(x) \rightarrow Bird(x) \wedge \neg CanFly(x) \\ \forall x : FlyingBird(x) \rightarrow Bird(x) \wedge CanFly(x) & \end{aligned}$$

We want to keep in the definition of the concept *Bird* (subsuming the unsatisfiable concept *Penguin*) a maximum of information that does not conflict with the definition of *Penguin*. The conflicting information is moved to the definition of the new concept *Flying bird*, which is declared to subsume all former subconcepts of *Bird* (such as *Canary* for example) except *Penguin*.

The algorithms developed in [23] and [24] are intended to detect problematic axioms causing a contradiction, to define the type of the contradiction (polysemy or overgeneralization) and to automatically repair the terminology by rewriting parts of the axioms that are responsible for the contradiction. Detected polysemous concepts are renamed and overgeneralized concepts are split into more general and more specific ones. Such solutions for a constant adaptation process of background knowledge are a first step towards a general theory of dynamification and adaptation of background knowledge. Although the framework has been tested in the area of text technology, it can be straightforwardly extended to a wider range of applications.

### 3.3 Neuro-Symbolic Learning Device

In order to bridge the gap between symbolic and sub-symbolic approaches, we sketch the theory presented in [11] based on the idea of translating first-order logical formulas into a variable-free representation in a topos [9]. A topos is a category theoretic structure consisting of objects *Obj* and arrows *Ar* having their domain and codomain in *Obj*. Certain construction principles allow to generate new arrows from old arrows. A fundamental theorem connects first-order logic and topos theory: a topos can be interpreted as a model of predicate logic [9]. The overall idea of learning symbolic theories with neural networks can be summarized as follows (compare also Figure 1):



**Fig. 1.** The architecture for learning a first-order logical theory with neural networks

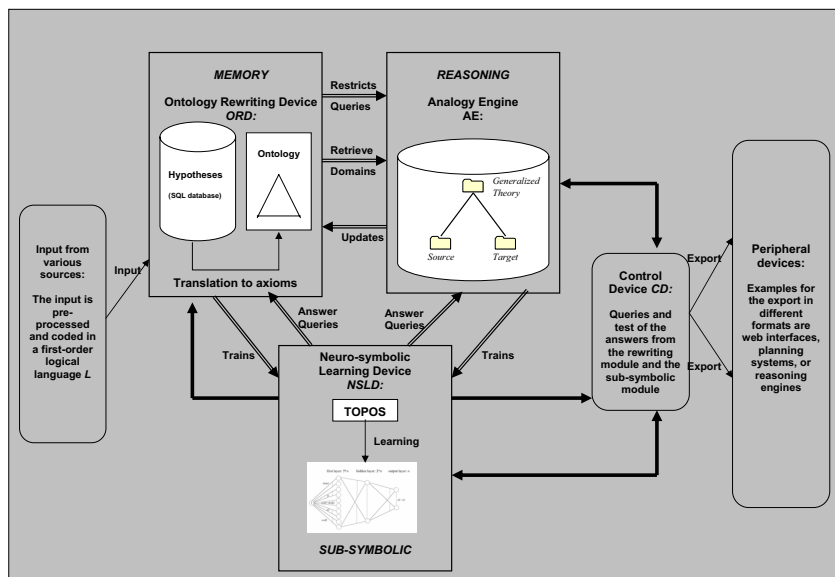
- First, input data is given by a set of logical formulas in a language  $\mathcal{L}$ .
- Second, this set of formulas is translated into objects and arrows of a topos. The representation is variable-free and homogeneous, i.e. only objects and arrows are represented combined by the operation concatenation of arrows.
- Third, a PROLOG program generates equations in normal form  $f \circ g = h$  identifying new arrows in the topos.
- Last but not least, these equations are used as input for the training of a neural network. The network has a standard feedforward topology and learns by backpropagation: the network adapts the representations of arrows in such a way that the arrows that correspond to “true logical expressions” are approximating the arrow *true*. The arrows *true* and *false* are the only hard-coded arrows.

Learning is possible, because the topos induces constructions that can be used for training. Although infinitely many constructions are induced by the topos, it turns out that a finite number is completely sufficient. We cannot go into details this approach. The interested reader is referred to [11] for more information. The framework has been tested with simple and also complex FOL theories.

## 4 The Integration of the Modules

### 4.1 A Hybrid Architecture for Higher Cognition

The three modules proposed in Section 3 – *NSLD*, *ORD*, and *AE* – attempt to learn a robust model of ontological background knowledge using a connectionist learning device, to dynamically rewrite ontologies on the symbolic level, and to perform various forms of reasoning, respectively. The integration of these modules can be achieved as follows: symbolic and sub-symbolic processes can be integrated, because *NSLD* is trained on symbolic data (i.e. on first-order logical



**Fig. 2.** The overall architecture for an integration of the different modules. Whereas the modules *ORD* and *NSLD* are adapting new ontological axioms to an existing ontology, the analogy engine *AE* computes analogical relations based on background knowledge provided by the other two modules. The control device *CD* is intended to choose answers from all three modules.

expressions) and it learns a model of a logical theory. Although it is currently not possible to extract directly symbolic information from *NSLD*, a competition of *ORD* and *NSLD* can be implemented by querying both and evaluating their answers. Furthermore both modules can directly interact due to the fact that input from *ORD* can be used for training and querying. A similar remark holds for the integration of *AE* and *NSLD*. Figure 2 depicts the overall architecture of the system.

- The input may originate from various sources, e.g. from resources based on structured data, unstructured data, or semi-structured data. The input needs to be available in an appropriate (subset) of a first-order language  $\mathcal{L}$ , in order to be in an appropriate format for the other modules. Therefore *ORD* generates appropriate logical formula from hypotheses.
- An important aspect is the interaction of *ORD* and *NSLD*: on the one hand, *ORD* trains *NSLD*, on the other hand *ORD* queries *NSLD*. Although *NSLD* can only give an approximate answer in terms of a classification, this can improve the performance of *ORD* in time-critical situations.
- With respect to the interaction of *AE* and *ORD*, ontological knowledge can naturally be used to constrain the computation of possible analogies [10]. Furthermore newly generated analogies can be used to update and therefore rewrite background knowledge [14].

- Similarly to the relation between *ORD* and *NSLD*, *AE* is used to train *NSLD*, whereas query answering can be performed in the other direction.
- The control device *CD* of the two learning modules is intended to implement a competition of the feedback of the three modules with respect to queries. Feedback may be in accordance to each other or not. In the second case, the ranking of the corresponding hypotheses is decided by *CD* (see below).

We exemplify the interaction between *AE* and *ORD* in more detail: the establishment of an analogical relation, if successful, provides a new conceptualization of the target domain. The metaphor in Subsection 3.1 results in a new conceptualization, where current is flowing in an electric circuit (triggered by a source). With respect to the ontology rewriting device *ORD* this means an update has to be performed, resulting in the introduction of a new (perhaps polysemous) concept, the update of a known concept using new relational constraints (flowing in an electric circuit), or even the generation of a conflict in the knowledge base (which has to be resolved). Additionally, the generalized theory of the anti-unification process introduces a new concept specifying an abstract circuit, where an entity is flowing caused by a source. On the other hand, *ORD* can be used to restrict possible analogical relations computed by *AE*: Due to the fact that *AE* can generalize arbitrary concepts, ontological knowledge may be used to restrict certain undesirable generalizations. For example, for a physics domain containing concepts like *time-point*, *real number*, *force*, *pressure* etc., it is undesirable to generalize *force* with *real number* or *pressure* with *time-point*. But it is desirable to generalize different types of *force*, or different types of *pressure*. Such restrictions can be implemented by specifying an upper-level ontology in *ORD* blocking certain (logically possible) generalizations.

We continue with some remarks concerning *CD*. This module evaluates possible answers of the three main modules and needs to implement a competition process. The natural way to realize such a control mechanism is to learn the behavior of the systems based on certain heuristics. We exemplify possible situations with respect to *ORD* and *NSLD*: in underdetermined situations, *ORD* is not able to answer queries, simply because *ORD* will not be able to prove anything without sufficient knowledge. In contrast to *ORD*, *NSLD* will be able to give an answer in any case. In such cases the usage of *NSLD* is clearly preferred by the heuristic. On the other hand, if *ORD* is able to prove a particular fact, for example, that a certain subsumption relation holds between two concepts *A* and *B*, then this result should be tentatively preferred by *CD* in comparison to the output of *NSLD*. In cases, in which time-critical reactions are necessary and *ORD* is not able to compute an answer in time, the natural heuristic would be to use *NSLD* instead. Finally, it could happen that the answers of *ORD* and *NSLD* are contradicting each other. In this case, *CD* cannot base the decision on *a priori* heuristics. A natural solution to this problem is to implement a reinforcement learning mechanism on *CD* itself, namely the learning of preferred choices (dependent on the particular domain) of the involved knowledge modules.

## 4.2 The Added-Value of a Hybrid Approach

The added-value of the overall architecture (as depicted in Figure 2) can be summarized as follows:

- The architecture is robust, because the trained neural network can give answers to queries, even though noise might be contained in the training data.
- Even in time-critical situations the proposed framework is able to react and to provide relevant information, because the neural network can answer a query immediately without any processing time, although the symbolic rewriting module may be busy with computation tasks.
- The architecture gives a first idea how an interaction between a symbolic level and a sub-symbolic level of computation can be achieved. The crucial issue is that *NSLD* is able to learn from highly structured training data.
- The architecture is cognitively more plausible than pure symbolic or sub-symbolic approaches. Although the hard problem of cognitive science, namely how a one-to-one translation from the symbolic level to the corresponding neural correlate and vice versa can be defined, is not reached yet, at least one direction of such a translation can be modeled.

Besides the mentioned advantages of such an architecture for automatically learning and adapting lexical ontologies, there is an important cognitive aspect that should be mentioned: phenomenologically the dualism between a conceptual level of cognition (*mind perspective*) and a neural level of cognition (*brain perspective*) is hardly questionable. From the *mind perspective*, cognition rests fundamentally on conceptual knowledge, i.e. on complex data structures, whereas from the *brain perspective* knowledge is not visible, but somehow hidden in the weights of the network or in specialized single neurons (or families of neurons) etc. To put it differently, although humans tend to use language as a prototypical tool for coding conceptual knowledge, it is not clear what the corresponding neural correlate should be, although this correlate is the only objectively measurable and directly accessible activity. Perhaps one can circumvent this hard problem of cognitive science by rooting higher cognitive abilities in models that humans use in order to think, to reason, or to communicate, instead of manipulating symbolic systems. If this is true, then this model of a conceptual theory (in our case of a logical first-order theory) can be coded on the neural level in a trained neural network. Additionally, this is complemented by a symbolic representation of semantic knowledge about the environment, allowing classical deductions and reasoning processes. In total, we think that the proposed hybrid architecture seems to be cognitively more plausible than isolated approaches that are purely based on one computational reasoning mechanism and representation paradigm.

## 5 Related Work

Some application domains for analogical reasoning were already mentioned in Section 3. With respect to the underlying methods for analogy making, algebraic

accounts [19], graph-based approaches [5], and similarity-based approaches [8] are to be mentioned.

A collection of approaches attempting to resolve inconsistencies in knowledge representation is related to classical methods of non-monotonic reasoning in logical systems. An example is the extension by default sets [16]. In [4], inductive logic programming techniques are proposed to resolve ontological inconsistencies. A family of approaches is based on tracing techniques for detecting a set of axioms that are responsible for particular contradictions in an ontology [20].

With respect to neuro-symbolic integration, we mention as examples sign propagation [22], tensor product representations [28], or holographic reduced representations [25]. Furthermore, researchers tried to model inferences with neural networks. An example to solve this problem is described in [17] in which a logical deduction operator is approximated by a neural network.

Recently, some endeavor has been invested to address the problem of achieving HLI. [3] proposes a so-called cognitive substrate in order to reduce higher cognition to a basis of low computational complexity. [6] propose to explain cognitive diversity as a reduction to the well-known structure-mapping theory. Another research tradition for higher cognition is the development of cognitive architectures like the AMBR/DUAL model [21] or the NARS architecture [30].

## 6 Conclusions and Future Research

The paper proposes a hybrid architecture, based on analogical reasoning, an ontology rewriting device, and a module for neuro-symbolic integration, in order to model HLI. Although each module has been proven to be successfully applicable in theory and practice to the respective domains, many challenges remain open. Besides the fact that the overall architecture needs to be implemented and carefully evaluated, there are several theoretical questions that need to be addressed. One aspect concerns the control architecture, in particular, the question on which basis competing answers from the different modules are evaluated. Another issue concerns the interaction of the particular modules: for example, whereas the training of the *NSLD* module by *ORD* is more or less well-understood, the other direction, i.e. the input from *NSLD* to *ORD* is (at present) rather unclear. Consequently, it is currently only possible to query the neural network, because a direct extraction of symbolic knowledge from the trained network is an unsolved problem. Additionally, the problem of the profusion of knowledge and representation formalisms needs to be addressed. It may be a possibility to restrict ontological knowledge practically to hierarchical sortal restrictions that can be coded by relatively weak description logics, but in the long run, this is probably not sufficient. Last but not least, it would be desirable to add further devices to the system, e.g. planning systems and action formalisms.



## Acknowledgment

This work has been partially supported by the German Research Foundation (DFG) through the grants KU 1949/2-1 and MO 386/3-4.

## References

1. Anderson, J., Thompson, R.: Use of analogy in a production system architecture. In: Vosniadou, O. (ed.) *Similarity and analogical reasoning*, Cambridge, pp. 267–297 (1989)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, D. (eds.): *Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge (2003)
3. Cassimatis, N.: A Cognitive Substrate for Achieving Human-Level Intelligence. *AI Magazine* 27(2), 45–56 (2006)
4. Fanizzi, N., Ferilli, S., Iannone, L., Palmisano, I., Semeraro, G.: Downward Refinement in the ALN Description Logic. In: *HIS 2004. Proc. of the Fourth International Conference on Hybrid Intelligent Systems*, pp. 68–73 (2004)
5. Falkenhainer, B., Forbus, K., Gentner, D.: The structure-mapping engine: Algorithm and example. *Artificial Intelligence* 41, 1–63 (1989)
6. Forbus, K., Hinrichs, T.: Companion Cognitive Systems: A step towards human-level AI. *AI Magazine* 27(2), 83–95 (2006)
7. Gentner, D.: Why We’re So Smart. In: Gentner, D., Goldin-Meadow, S. (eds.) *Language in mind: Advances in the study of language and thought*, pp. 195–235. MIT Press, Cambridge (2003)
8. Gentner, D.: The mechanisms of analogical learning. In: Vosniadou, S., Ortony, A. (eds.) *Similarity and Analogical Reasoning*, pp. 199–241. Cambridge University Press, New York (1989)
9. Goldblatt, R.: *Topoi: The Categorical Analysis of Logic*. In: *Studies in Logic and the Foundations of Mathematics*, vol. 98, North-Holland, Amsterdam (1979)
10. Gust, H., Kühnberger, K.-U., Schmid, U.: Ontological Aspects of Computing Analogies. In: *Proceedings of the 6th International Conference on Cognitive Modeling*, pp. 350–351. Lawrence Erlbaum, Mahwah (2004)
11. Gust, H., Kühnberger, K.-U.: Learning Symbolic Inferences with Neural Networks. In: Bara, B., Barsalou, L., Bucciarelli, M. (eds.) *CogSci 2005, XXVII Annual Conference of the Cognitive Science Society*, pp. 875–880. Lawrence Erlbaum, Mahwah (2005)
12. Gust, H., Kühnberger, K.-U.: Explaining Effective Learning by Analogical Reasoning. In: Sun, R., Miyake, N. (eds.) *CogSci/ICCS 2006. 28th Annual Conference of the Cognitive Science Society*, pp. 1417–1422. Lawrence Erlbaum, Mahwah (2006)
13. Gust, H., Kühnberger, K.-U., Schmid, U.: Metaphors and Heuristic-Driven Theory Projection (HDTP). *Theoretical Computer Science* 354, 98–117 (2006)
14. Gust, H., Kühnberger, K.-U., Schmid, U.: Ontologies as a Cue for the Metaphorical Meaning of Technical Concepts. In: Schalley, A., Khlentzos, D. (eds.) *Mental States: Evolution, Function, Nature*, pp. 191–212. John Benjamins Publishing Company, Amsterdam (to appear)
15. Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A framework for handling inconsistency in changing ontologies. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) *ISWC 2005. LNCS*, vol. 3729, Springer, Heidelberg (2005)

16. Heymans, S., Vermeir, D.: A Defeasible Ontology Language. In: Meersman, R., Tari, Z., et al. (eds.) CoopIS 2002, DOA 2002, and ODBASE 2002. LNCS, vol. 2519, Springer, Heidelberg (2002)
17. Hitzler, P., Hölldobler, S., Seda, A.: Logic programs and connectionist networks. *Journal of Applied Logic* 2(3), 245–272 (2004)
18. Hofstadter, D.: The Fluid Analogies Research Group: Fluid concepts and creative analogies. Basic Books, New York (1995)
19. Indurkha, B.: *Metaphor and Cognition*. Kluwer, Dordrecht (1992)
20. Kalyanpur, A.: *Debugging and Repair of OWL Ontologies*. Ph.D. Dissertation (2006)
21. Kokinov, B., Petrov, A.: Integrating Memory and Reasoning in Analogy-Making: The AMBR Model. In: Gentner, D., Holyoak, K., Kokinov, B. (eds.) *The Analogical Mind. Perspectives from Cognitive Science*, Cambridge Mass (2001)
22. Lange, T., Dyer, M.G.: High-level inferencing in a connectionist network. Technical report UCLA-AI-89-12 (1989)
23. Ovchinnikova, E., Kühnberger, K.-U.: Adaptive  $\mathcal{AL}\mathcal{E}$ -TBox for Extending Terminological Knowledge. In: Sattar, A., Kang, B.-H. (eds.) *AI 2006*. LNCS (LNAI), vol. 4304, pp. 1111–1115. Springer, Heidelberg (2006)
24. Ovchinnikova, E., Wandmacher, T., Kühnberger, K.U.: Solving Terminological Inconsistency Problems in Ontology Design. *International Journal of Interoperability in Business Information Systems* 2(1), 65–80 (2007)
25. Plate, T.: *Distributed Representations and Nested Compositional Structure*. PhD thesis, University of Toronto (1994)
26. Plotkin, G.: A note of inductive generalization. *Machine Intelligence* 5, 153–163 (1970)
27. Schwering, A., Krumnack, U., Kühnberger, K.-U., Gust, H.: Using Gestalt Principles to Compute Analogies of Geometric Figures. In: McNamara, D.S., Trafton, J.G. (eds.) *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, pp. 1485–1490. Cognitive Science Society, Austin, TX (2007)
28. Smolenski, P.: Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence* 46(1–2), 159–216 (1996)
29. Staab, S., Studer, R. (eds.): *Handbook of Ontologies*. Springer, Heidelberg (2004)
30. Wang, P.: *Rigid Flexibility: The Logic of Intelligence*. Springer, Heidelberg (2006)

# A Rule-Based System for Assessing Consistency Between UML Models

Carlos Mario Zapata<sup>1</sup>, Guillermo González<sup>1</sup>, and Alexander Gelbukh<sup>2</sup>

<sup>1</sup> Escuela de Sistemas, Universidad Nacional de Colombia,  
Carrera 80 N° 65-23, Bloque M8. Medellín, Colombia  
Tel.: 4255350

{cmzapata, ggonzal}@unalmed.edu.co

<sup>2</sup> Computing Research Center (CIC), National Polytechnic Institute (IPN),  
Col. Zacatenco, 07738, DF, Mexico  
www.Gelbukh.com

**Abstract.** The main goal of requirements specification is the transformation of a “rough draft” of stakeholder needs and expectations into a semi-formal specification, represented by several diagrams, commonly UML diagrams. These diagrams must be consistent with each other, but consistency among different UML diagrams is not defined by the UML specification, and the research about inter-model consistency is still immature. We propose, in this paper, a rule-based system to detect consistency problems among UML diagrams. In order to complete this task, we have defined a set of rules in OCL, and then we use a novel approach for implementing the system by means of Xquery and Xpath languages. The use of these languages helps the rule-based system to interact with traditional CASE tools.

## 1 Introduction

The initial specification of a software application is often informal and possibly vague, and it is usually a “rough draft” of the final specification [1]. This commonly incomplete and inconsistent “rough draft” must be translated to a correct requirement specification, and then presented to the stakeholders for their validation. One of the critical tasks in requirements engineering tries to assure the quality of the step-by-step specification, in order to find consistency, correctness, and completeness mistakes as soon as possible in the software lifecycle [2]. The Unified Modeling Language (UML) is often used to make such specification [3].

Consistency has been, particularly, one of the most important concerns of software development process, and there are lots of works about it. However, there are still problems to be solved:

- UML superstructure [3] and other works [4] only define intra-model consistency. The inter-model consistency is not formally specified [5] and it is not supported by the common CASE (Computer-Aided Software Engineering) tools.
- Consistency checking is carried out automatically among some of the models and the executable code [6]. Executable code is the final step in the software

development lifecycle, and we need to perform consistency checking in the previous stages (definition, analysis, and design).

- Several works [7, 8, 9, and 10] establish transformation processes instead of consistency checking processes. We need to specify the way consistency checking is performed among diagrams, and transformation processes do not help in such task.
- Some works [4 and 10] are done in a semi-automated way; if the analyst must participate in the consistency checking process, this process will probably be human-error-prone.
- There is one approach to specify consistency checking in a semi-formal way [11]. Lack of formalism can cause ambiguity problems in the final specification of a software application.

The reviewed works use rules in order to define the consistency checking process. In essence, rules are the main elements of the rule-based systems, a contribution of the Artificial Intelligence (AI) to the solution of this kind of problems. Ligêza [12] defined a set of principles for designing rule-based systems, and one of the most important is functional capability, a principle linked to the functionality of the language in which the rule-based system is programmed. In reference to the problem of consistency checking, we need to guarantee that the models, commonly made in CASE tools, can be accessed by the rule-based system in order to check consistency among them. Because of this, we need to introduce XML (Extended Markup Language) capabilities in the rule-based system.

In this paper we propose a method for verifying consistency between UML diagrams by means of a novel approach to rule-based systems. The rules of the system are defined in OCL [3], a formal language for constraint definition, and then they are programmed in Xpath and Xquery, special languages for selecting and processing parts of XML code. For sake of exemplification, the rules are related to consistency between class and use case diagrams, two of the most important diagrams of UML.

The structure of this paper is as follows: in Section 2 we review specialized literature about consistency checking. Section 3 describes Xpath and Xquery and justifies the use of these languages in the rule-based system. Section 4 presents the rule-based system for consistency checking. Finally, in Section 5 the conclusions and future work are given.

## 2 Literature Overview on Consistency Checking in UML Diagrams

The main source of consistency rules for UML diagrams is the UML superstructure, emitted by the OMG [3]. This document includes some intra-model consistency rules in OCL (Object Constraint Language), a formal language for constraint specification; some of these rules are implemented in some of the conventional CASE tools. However, inter-model consistency rules are not defined. Some works have been developed [4–11] for dealing with the problem of consistency between different kinds of models.

Dan Chiorean [4] used OCL [3] for checking the intra-model consistency in UML diagrams, with the help of the CASE tool OCLE. This tool is compatible with XMI

[13], and it can process UML models generated by many of the available CASE tools (Together, Rational Rose, MagicDraw, Poseidon, ArgoUML, etc.). A software specification is integrated by many diagrams, and intra-model consistency checking is only a part of the job; in order to guarantee the quality of a complete specification, we need to check consistency between several UML diagrams.

Xlinkit [6] is an environment for consistency checking of heterogeneous distributed documents. This approach uses several languages like XML, Xpath, Xlink, and DOM [13], and is conformed by a first-order-logic-based language to express constraints among documents, a document management system, and an engine for checking the constraints against the documents. In the runtime, Xlinkit applies Xpath expressions in order to review all the documents of the collection, and then it constructs a list of nodes to be checked. The documents can be referred to UML class diagrams or to source code of the software application. However, such comparison does not make sense; the source code is available in the implementation stage of software lifecycle, and we need to discover potential mistakes in the specification in previous stages.

Four parallel works: Kösters, Pagel and Winter [7], Liu *et al.* [8], Shishkov *et al.* [9], and Buhr [10], use case diagram to derive the class diagram. Transformation between diagrams is a way to guarantee the consistency between diagrams, but only in the case that we can completely generate the second diagram from the first. This is not the common case of software specification, where every diagram contains both proper information and shared information. In other words, by means of the transformation process, we only can generate the shared information of the second diagram, and the independent information must be completed in a manual process.

Glinz [5] defines a manual method for assessing consistency between class and use case diagrams, and he uses as a starting point a textual specification of the use cases in a special format. In this method, the presence of the analyst is highly required in the consistency checking process among the two diagrams, and this situation makes it difficult for the partial or total automation of the process.

Sunetnanta and Finkelstein [11] present an approach for checking the inter-model consistency, and they based this approach on the UML diagram conversion into conceptual graphs, and on the definition of consistency rules referred to the same graphs. The conceptual graphs can not be considered as a formal approach for this kind of specification elaboration, but a semi-formal approach. While the approaches have still low formal level, there is a high probability of ambiguity problems in the consistency checking process.

### 3 Xpath, Xquery, and Rule-Based Systems

XML is an extremely versatile markup language, capable of labeling the information content of diverse data sources, including structured and semi-structured documents, relational databases, and object repositories. Throughout the last few years, the use of XML [14] has grown, and this language has become a standard language for communication purposes among applications. The reasons why the use of this language has increased are the strange mix of suitability and standardization that it can achieve. Also, XML has an important suite of standard languages surrounding it,

and this suite gives it the power to interact among different applications. Two of the most important languages of this suite are Xquery and Xpath [13].

A query language that uses the structure of XML can intelligently express queries across all these kinds of data, whether physically stored in XML, or viewed as XML via middleware. Query languages have been traditionally designed for specific kinds of data. Existing proposals for XML query languages are robust for particular types of data sources, but weak for other types. The Xquery specification has been designed to be broadly applicable across all types of XML data sources. Xquery is a functional language to acquire data in multiple document formats (including XML documents) and then to produce XML-based results [13]. Xquery extensively uses the so-called Xpath, an expression language to select parts of an XML document by means of a matching process [13]. Both Xquery and Xpath languages are used to retrieve information pieces, especially from XML documents; for this reason, these languages have been commonly used for processing information in the semantic web.

The growing use of XML-based languages has motivated the extension of some of their capabilities, and rule-based systems have been employed for this purpose, especially in the fields of query optimization [15, 16, and 17] and logic programming [18]. By contrast, Xquery and Xpath can be used to support the elaboration of rule-based systems, as suggested by Eguchi and Leff [19], who discussed the use of XML-based languages to create artificial intelligence applications in the legal environment. Following the same trend, Schaffert [20] proposed a special rule-based language called Xcerpt, which is suitable to create rule-based systems from web documents.

The UML superstructure [3] suggests XMI (XML Metadata Interchange) as a standard language to share information among UML-based applications. Most of the UML-based CASE tools are capable to export diagrams in XMI format, for the purpose of communication among them. In order to create a rule-based system to check consistency between UML diagrams, the previous approaches do not use XMI; only Xlinkit [6] uses a sort of XML-based environment, but the goal of this environment is the comparison of UML diagrams against source code. Due to the fact that Xcerpt [20] is suitable for the semantic web, it does not use XMI as a way to interchange information. By this reason, in the next section we propose the use of Xquery and Xpath for creating a rule-based system to check consistency between two of the most common UML diagrams: class and use case diagrams.

## 4 A Rule-Based System for Consistency Checking Between UML Models

The construction of a rule-based system to define the consistency rules between ML class and use case diagrams requires the definition of some principles:

- Every class is distinguished by its name, by a collection of properties, and by a collection of operations offered by the class.
- The use case model has actors, which represents the roles which different users can play, and use cases, which represents the actions performed by the actors in the future software application.

Due to the fact that the UML superstructure [3] only defined intra-model rules, a heuristic analysis of the experience of software analysts was performed to define a set of consistency rules between class and use case diagrams. In this section, we present and specify two of such rules; the rules are presented in natural language, OCL, and Xquery-Xpath source code.

**Rule 1.** The name of a use case must include a verb and a noun; the noun should correspond to the name of one class in the class diagram. In other words, for each use case U in the class diagram, there should be a class C belonging to the class diagram, so that  $U.name$  equals  $C.name$ . Figure 1 depicts the graphical representation of this rule.

The OCL expression that represents this rule is shown in Figure 1.

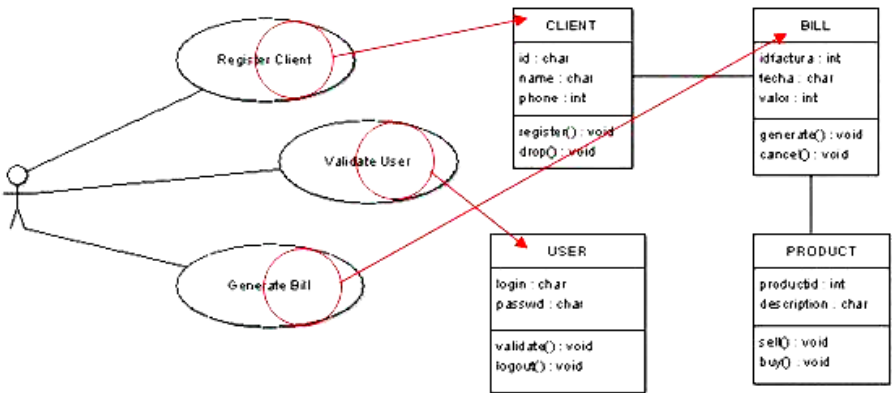


Fig. 1. Graphical expression of Rule 1

Classifier

*self.UseCase->exists(us: Usecase, c: Class, x: Integer, y: Integer | y > x and us.name.toUpper.substring(x,y)=c.name.toUpper)*

The Xquery-Xpath expression that represents this rule is:

```
<rule1>{
  for $i in 1 to count($class)
  for $j in 1 to count($usecase)
  return
    if (contains(upper-case($usecase[position()=$j]/@name), upper-
      case($class[position()=$i]/@name))) then
      ("<br/>The class <b> ", upper-
        case($class[position()=$i]/@name), "</b> exists in the use case
        <b>", upper-case($casouso[position()=$j]/@name), "</b>")
```

else

("<br/>The class <b> ", upper-case(\$clase[position()=\$i]/@name), "</b> not exists in the use case <b>", upper-case(\$casouso[position()=\$j]/@name), "</b>")

</rule1>

**Rule 2.** The name of a use case must include a verb and a noun; the verb should correspond to an operation of a class in the class diagram that was identified in rule 1. In other words, for each use case U there should be a class C that contains an operation *Operationx* so that *U.name* contains *C.Operationx*. Figure 2 depicts the graphical representation of this rule.

The OCL expression that represents this rule is shown in Figure 2.

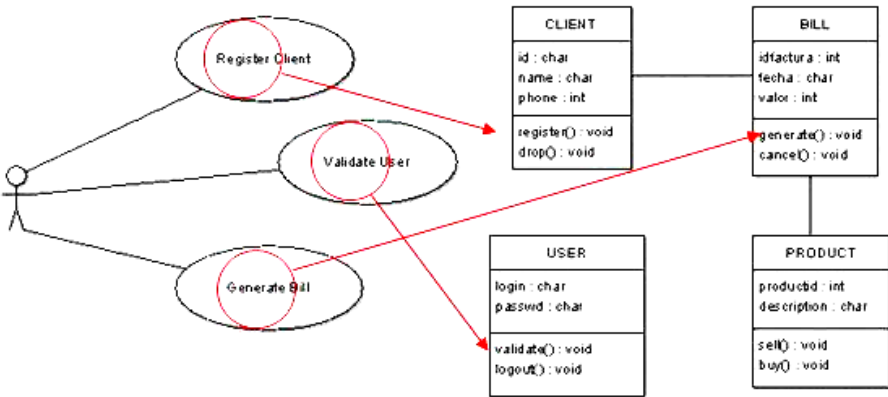


Fig. 2. Graphical expression of Rule 2

Classifier

self.UseCase->exists(us: Usecase, c: Class, x: Integer, y: Integer | y > x and us.name.toUpper.substring(x,y)=c.operation.toUpper)

The Xquery-Xpath expression that represents this rule is:

```
<rule2>{
  for $i in 1 to count($operation)
  for $j in 1 to count($usecase)
  return
    if (contains(upper-case($usecase[position()=$j]/@name), upper-
      case($operation[position()=$i]/@name))) then
      ("<br/>The operation <b>", upper-
        case($operation[position()=$i]/@name), "</b> exists in the use case
        <b>", upper-case($usecase[position()=$j]/@name), "</b>")
```



*else*

```
("<br/>The operation <b>",upper-
case($operation[position()=$i]/@name), "</b> not exists in the use
case <b>",upper-case($usecase[position()=$j]/@name), "</b>" )
```

```
}</rule2>
```

The complete set of rules was programmed in a rule-based system for checking consistency between class and use case diagrams. The inputs of the system are the two diagrams in XMI format [3]; ArgoUML® was the CASE tool selected to make these diagrams, and then to export them to XMI. The rule-based system was programmed in Java®, and uses Xquery and Xpath languages by means of an API (Application Program Interface) named *Saxon* for the validation of the rules. The rule-based system assesses the diagrams and creates a report to inform if the rules are followed (correct state), or are not (error state). Also, the rule-based system informs if there is an error of synonyms; to achieve this goal, the system uses a word list, which includes possible synonyms of every word. When the system detects two synonyms used in the same diagram, a warning message is presented.

If we apply the described process to the diagrams of Figure 1, after we introduce the XMI file resulting from ArgoUML®, in the rule-based system we achieve a XML file with the information of the recognized class diagram (see, for example, the class “BILL” in Figure 3; all of the classes will exhibit the same appearance), use case diagram (see Figure 4), and the results of consistency checking process (see Figure 5).

```
<class>
<name>BILL</name>
<attribute>ID</attribute>
<attribute>DATE</attribute>
<attribute>AMOUNT</attribute>
<operation>GENERATE</operation>
<operation>CANCEL</operation>
</class>
```

Fig. 3. XML file corresponding to the class “BILL”

```
<usecase>
<name>REGISTER CLIENT</name>
</usecase>
<usecase>
<name>VALIDATE USER</name>
</usecase>
<usecase>
<name>GENERATE BILL</name>
</usecase>
```

Fig. 4. XML file corresponding to the use case diagram

```

<consistency>
<existsclassinusecase>
The class BILL exists in the use case GENERATE BILL
The class USER exists in the use case VALIDATE USER
The class CLIENT exists in the use case REGISTER CLIENT
The class PRODUCT not exists in the use case REGISTER CLIENT
The class PRODUCT not exists in the use case VALIDATE USER
The class PRODUCT not exists in the use case GENERATE BILL
</existsclassinusecase>

<existsoperationinusecase>
The operation GENERATE exists in the use case GENERATE BILL
The operation CANCEL not exists in the use case GENERATE BILL
The operation REGISTER exists in the use case REGISTER CLIENT
The operation DROP not exists in the use case REGISTER CLIENT
The operation VALIDATE exists in the use case VALIDATE USER
The operation LOGOUT not exists in the use case VALIDATE USER
The operation SELL not exists in the use case REGISTER CLIENT
The operation SELL not exists in the use case VALIDATE USER
The operation SELL not exists in the use case GENERATE BILL
The operation BUY not exists in the use case REGISTER CLIENT
The operation BUY not exists in the use case VALIDATE USER
The operation BUY not exists in the use case GENERATE BILL
</existsoperationinusecase>
</consistency>

```

Fig. 5. XML file corresponding to the consistency checking process

## 5 Conclusions and Future Work

A novel approach to use Xquery and Xpath in the development of a rule-based system was presented in this paper. The main goal of the system is the assessment of consistency rules between UML class and use case diagrams.

This work makes contributions to Requirements Engineering and Artificial Intelligence. In the first case, the problem of definition of inter-model rules for consistency checking was dealt with by means of a formal specification of consistency rules between the use case and class diagrams in OCL. With the integration of the OCL in the rules definition, we assure that there is a formal way to check them, in order to avoid ambiguities and to guarantee well formed models. As a future work, a possible integration with the well-formedness rules of the UML specification can be defined. Related to Artificial Intelligence, this work has showed a novel way to incorporate XML-based languages in the development of rule-based systems. XML technology facilitates the access to several sources of information (for example, the semantic web and, in this particular situation, to the diagrams made by

means of CASE tools) and, in conjunction with the Artificial Intelligence theory, becomes a better way to develop rule-based systems.

Additional future work must be directed to extend the rule-based system to other UML diagrams (for example activity and sequence diagrams) and to other requirements engineering diagrams (for example goal diagram and process diagram). Also, we need to examine languages like Xcerpt, for assessing the suitability of these approaches to rule-based systems, with the possibility of accessing the diagrams made by means of many CASE tools.

**Acknowledgements.** The work was done with partial support of Mexican Government (SNI, CONACYT, COFAA-IPN) to the third author.

## References

1. Jackson, M.: *Software Requirements & Specifications: a lexicon of practice, principles and prejudices*. Addison Wesley, Great Britain (1995)
2. Zowghi, D., Gervasi, V.: The Three Cs of requirements: consistency, completeness, and correctness. In: *International Workshop on Requirements Engineering: Foundations for Software Quality*, Essen, pp. 155–164. Essener Informatik Beitiage, Germany (2002)
3. OMG – Object Management Group. <http://www.omg.org>
4. Chiorean, D., Pasca, M., Carcu, A., Botiza, C., Moldovan, S.: Ensuring UML models consistency using the OCL Environment. In: *Sixth International Conference on the Unified Modelling Language - the Language and its applications*, San Francisco (2003)
5. Glinz, M.: A lightweight approach to consistency of Scenarios and Class Models. In: *En: Fourth International Conference on Requirements Engineering*, Illinois, USA, June 10-23 (2000)
6. Gryce, C., Finkelstein, A., Nentwich, C.: Lightweight Checking for UML Based Software Development. In: *Workshop on Consistency Problems in UML-based Software Development*. Dresden, Germany (2002)
7. Kösters, G., Pagel, B.-U., Winter, M.: Coupling Use Cases and Class Models. In: *BCS FACS/EROS Workshop on Making Object-oriented Methods more Rigorous*, London (1997)
8. Liu, D., Subramaniam, K., Far, B.H., Eberlein, A.: Automating transition from use-cases to class model. In: *IEEE CCECE 2003. Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 831–834 (2003)
9. Shishkov, B., Xie, Z., Lui, K., Dietz, J.: Using norm analysis to derive use case from business processes. In: *5th Workshop on Organizations semiotics*, Delft the Netherlands, June 14-15, 2002 (2002)
10. Buhr, R.J.A.: Use Case Maps as Architectural Entities for Complex Systems. *IEEE Transactions on Software Engineering* 24(12), 1131–1155 (1998)
11. Sunetnanta, T., Finkelstein, A.y.: Automated Consistency Checking for Multiperspective Software Specifications. In: *Proceedings of the 26th Australasian computer science conference*, vol. 16, pp. 291–300 (2003)
12. Ligêza, A.: *Logical Foundations of Rule-Based Systems*. Studies in Computational Intelligence (SCI) 11, 191–198 (2006)
13. W3C – World Wide Web Consortium. <http://www.w3.org>
14. XML – Extensible Markup Language. <http://www.w3.org/XML>

15. Travers, N., Dang, T.: An Extensible Rule Transformation Model for XQuery Optimization. In: ICEIS. International Conference on Enterprise Information Systems, INSTICC, Madeira (2007)
16. Pal, S., Istvan, C., Seeliger, O., Rys, M., Schaller, G., Yu, W., Tomic, D., Baras, A., Berg, B., Churin, D., Kogan, E.: XQuery implementation in a relational database system. In: Proceedings of the 31st international conference on Very large data bases, pp. 1175–1186. Trondheim, Norway (2005)
17. Che, D., Aberer, K., Özsu, M.: Query optimization in XML structured-document databases. *The VLDB Journal* 15(3), 263–289 (2006)
18. Almendros, J., Becerra, A., Enciso, F.: Magic Sets for the XPath Language. *Journal of Universal Computer Science* 12(11), 1651–1678 (2006)
19. Eguchi, G., Leff, L.: Rule-based XML: Rules about XML in XML To Support Litigation Regarding Contracts. *Artificial Intelligence and Law* 10, 283–294 (2002)
20. Schaffert, S., Xcerpt, A.: Rule-Based Query and Transformation Language for the Web. PhD thesis, University of Munich (2004)

# Partial Satisfiability-Based Merging

Pilar Pozos Parra<sup>1</sup> and Verónica Borja Macías<sup>2</sup>

<sup>1</sup> Universidad Juárez Autónoma de Tabasco  
Carretera Cunduacán - Jalpa Km. 1 Cunduacán Tabasco, México  
pilar.pozos@dais.ujat.mx

<sup>2</sup> Universidad Tecnológica de la Mixteca  
Carretera a Acatlima Km 2.5 Huajuapán de León Oaxaca, México  
vero0304@mixteco.utm.mx

**Abstract.** When information comes from different sources inconsistent beliefs may appear. To handle inconsistency, several model-based belief merging operators have been proposed. Starting from different belief bases which might conflict, these operators return a unique consistent base which represents the beliefs of the group. The operators, parameterized by a distance between interpretations and aggregation function, usually only take into account consistent bases, consequently some information which is not responsible for conflicts may be ignored. An alternative way of merging uses the notion of Partial Satisfiability to define *PS-Merge*, a model-based merging operator that produces similar results to other merging approaches, but while other approaches require many merging operators in order to achieve satisfactory results for different scenarios *PS-Merge* obtains similar results for all these different scenarios with a unique operator. This paper analyzes some of the properties satisfied by *PS-Merge*.

## 1 Introduction

Belief merging is concerned with the process of combining the information contained in a set of (possibly inconsistent) belief bases obtained from different sources to produce a single consistent belief base. Belief merging is an important issue in artificial intelligence and databases, and its applications are many and diverse [1]. For example, in multiagent systems a merging operator defines the beliefs of a group of agents according to the beliefs of each member of the group. When agents have conflicting beliefs about the “true” state of the world, belief merging can be used to determine the “true” state of the world for the group. Though we consider only belief bases, merging operators can typically be used for merging either beliefs or goals.

Several merging operators have been defined and characterized in a logical way. Among them, model-based merging operators [2,3,4,5] obtain a belief base from a set of interpretations with the help of a distance measure on interpretations and an aggregation function. Usually model-based merging operators only take into account consistent belief bases consequently some information which is not responsible for conflicts may be ignored. Other merging operators,

syntax-based ones [6], are based on the selection of some consistent subsets of the set-theoretic union of the belief bases. This allows for taking inconsistent belief bases into account, but such operators usually do not take into account the frequency of each explicit item of belief. For example, the fact that a formula  $\psi$  is believed in a base or in  $n$  bases is not considered relevant, which is counter-intuitive.

An alternative way of merging uses the notion of Partial Satisfiability to define *PS-Merge*, a model-based merging operator which depends on the syntax of the belief bases [7]. The proposal produces similar results to other merging approaches, but while other approaches require many merging operators in order to achieve satisfactory results for different scenarios the proposal obtains similar results for all these different scenarios with a unique operator. It is worth noting that *PS-Merge* is not based on distance measures, and takes into account inconsistent bases and the frequency of each explicit item of belief. We study some logical properties satisfied by *PS-Merge* and analyze the rational behavior of the operator.

The rest of the paper is organized as follows. After providing some technical preliminaries, Section 3 describes the notion of Partial Satisfiability and the associated merging operator. Section 4 studies some properties satisfied by *PS-Merge* in the context of postulates proposed in [3,8]. In Section 5 we give a comparison of *PS-Merge* with other approaches and Section 6 concludes with a discussion of future work.

## 2 Preliminaries

We consider a language  $\mathcal{L}$  of propositional logic formed from  $P := \{p_1, p_2, \dots, p_n\}$  (a finite ordered set of atoms) in the usual way. And we use the standard terminology of propositional logic except for the definitions given below. A *belief base*  $K$  is a finite set of propositional formulas of  $\mathcal{L}$  representing the beliefs of the agent (we identify  $K$  with the conjunction of its elements).

A *state* or *interpretation* is a function  $w$  from  $P$  to  $\{1, 0\}$ , these values are identified with the classical truth values *true* and *false* respectively. The set of all possible states will be denoted as  $\mathcal{W}$  and its elements will be denoted by vectors of the form  $(w(p_1), \dots, w(p_n))$ . A *model* of a propositional formula  $Q$  is a state such that  $w(Q) = 1$  once  $w$  is extended in the usual way over the connectives. For convenience, if  $Q$  is a propositional formula or a set of propositional formulas then  $\mathcal{P}(Q)$  denotes the set of atoms appearing in  $Q$ .  $|P|$  denotes the cardinality of set  $P$ . A *literal* is an atom or its negation.

A *belief profile*  $E$  denotes the beliefs of agents  $K_1, \dots, K_m$  that are involved in the merging process,  $E = \{\{Q_{1_1}, \dots, Q_{n_1}\}, \dots, \{Q_{1_m}, \dots, Q_{n_m}\}\}$  where  $Q_{1_i}, \dots, Q_{n_i}$  denotes the beliefs in the base  $K_i$ .  $E$  is a multiset (bag) of belief bases and thus two agents are allowed to exhibit identical bases.

Two belief profiles  $E_1$  and  $E_2$  are said to be equivalent, denoted by  $E_1 \equiv E_2$ , iff there is a bijection  $g$  from  $E_1$  to  $E_2$  such that  $K \equiv g(K)$  for every base  $K$  in  $E_1$ . With  $\bigwedge E$  we denote the conjunction of the belief bases  $K_i \in E$ , while  $\sqcup$

denotes the multiset union. For every belief profile  $E$  and positive integer  $n$ ,  $E^n$  denotes the multiset union of  $n$  times  $E$ .

### 3 Partial Satisfiability

In order to define Partial Satisfiability without loss of generality we consider a normalized language so that each belief base is taken as the disjunctive normal form (DNF) of the conjunction of its elements. Thus if  $K = \{Q_1, \dots, Q_n\}$  is a belief base we will identify this base with  $Q_K = DNF(Q_1 \wedge \dots \wedge Q_n)$ . The DNF of a formula is obtained by replacing  $A \leftrightarrow B$  and  $A \rightarrow B$  by  $(\neg A \vee B) \wedge (\neg B \vee A)$  and  $\neg A \vee B$  respectively, applying De Morgan's laws, using the distributivity law, distributing  $\vee$  over  $\wedge$  and finally eliminating the literals repeated in each conjunct. The last part of the construction of the DNF (the minimization by eliminating literals) is important since the number of literals in each conjunct affects the satisfaction degree of the conjunct. We are not applying other logic minimization methods to reduce the size of the DNF expressions since this may affect the intuitive meaning of the formulas. An further analysis of logic equivalence and the results obtained by the Partial Satisfiability is required.

*Example 1.* Given the belief base  $K = \{a \rightarrow b, \neg c\}$  it is identified with  $Q_K = (\neg a \wedge \neg c) \vee (b \wedge \neg c)$ .

**Definition 1 (Partial Satisfiability).** Let  $K$  be a belief base,  $w$  any state of  $\mathcal{W}$  and  $|P| = n$ , we define the Partial Satisfiability of  $K$  for  $w$ , denoted as  $w_{ps}(Q_K)$ , as follows.

– If  $Q_K := C_1 \wedge \dots \wedge C_s$  where  $C_i$  are literals then

$$w_{ps}(Q_K) = \max \left\{ \sum_{i=1}^s \frac{w(C_i)}{s}, \frac{n - |\mathcal{P}(\bigwedge_{i=1}^s C_i)|}{2n} \right\}$$

– If  $Q_K := D_1 \vee \dots \vee D_r$  where each  $D_i$  is a literal or a conjunction of literals then

$$w_{ps}(Q_K) = \max \{w_{ps}(D_1), \dots, w_{ps}(D_r)\}$$

The intuitive interpretation of Partial Satisfiability is as follows: it is natural to think that if we have the conjunction of two literals and just one is satisfied then we are satisfying 50% of the conjunction. If we generalize this idea we can measure the satisfaction of a conjunction of one or more literals as the sum of the evaluation of them under the interpretation divided by the number of conjuncts. When the agent's beliefs consider only some atoms of the language, it is not affected by the decision taken over the atoms not appearing in its beliefs. Hence it is indifferent to the evaluation of these atoms, so we interpret this indifference as a partial satisfaction of 50% for each atom not appearing in its beliefs.

On the other hand the agent is interested in satisfying the literals that appear in its beliefs and we interpret this fact by assigning a satisfaction of 100% to

each literal verified by the state and 0% to those that are falsified. As we can see the former intuitive idea is reflected in Definition 1 since the literals that appear in the agents beliefs have their classical value and atoms not appearing have a value of just  $\frac{1}{2}$ . We will consider literals that do not appear just in case the satisfaction of the conjunction of the appearing literals is lower than the satisfaction obtained for those not appearing.

*Example 2.* The Partial Satisfiability of the belief base of Example 1 given  $P = \{a, b, c\}$  and  $w = (1, 1, 1)$  is

$$w_{ps}(Q_K) = \max \left\{ \max \left\{ \frac{w(\neg a) + w(\neg c)}{2}, \frac{1}{6} \right\}, \max \left\{ \frac{w(b) + w(\neg c)}{2}, \frac{1}{6} \right\} \right\} = \frac{1}{2}.$$

Instead of using distance measures [3,5,8,9] we have proposed the notion of Partial Satisfiability in order to define a new merging operator. The elected states of the merge are those whose values maximize the sum of the Partial Satisfiability of the bases.

**Definition 2.** Let  $E$  be a belief profile obtained from the belief bases  $K_1, \dots, K_m$ , then the Partial Satisfiability Merge of  $E$  denoted by  $PS\text{-Merge}(E)$  is a mapping from the belief profiles to belief bases such that the set of models of the resulting base is:

$$\left\{ w \in \mathcal{W} \mid \sum_{i=1}^m w_{ps}(Q_{K_i}) \geq \sum_{i=1}^m w'_{ps}(Q_{K_i}) \text{ for all } w' \in \mathcal{W} \right\}$$

*Example 3.* Revez in [10] proposes the following scenario. A teacher asks three students which among three languages, SQL, Datalog and  $O_2$ , they would like to learn. Let  $s, d$  and  $o$  be the propositional letters used to denote the desire to learn SQL, Datalog and  $O_2$ , respectively, then  $P = \{s, d, o\}$ . The first student only wants to learn SQL or  $O_2$ , the second wants to learn only one of Datalog or  $O_2$ , and the third wants to learn all three languages. So we have  $E = \{K_1, K_2, K_3\}$  with  $K_1 = \{(s \vee o) \wedge \neg d\}$ ,  $K_2 = \{(\neg s \wedge d \wedge \neg o) \vee (\neg s \wedge \neg d \wedge o)\}$ , and  $K_3 = \{s \wedge d \wedge o\}$ .

In [9] using the Hamming distance applied to the anonymous aggregation function  $\Sigma$  and in [3] using the operator  $\Delta_\Sigma$ , both approaches obtain the states  $(0, 0, 1)$  and  $(1, 0, 1)$  as models of the merging.

We have  $Q_{K_1} = (s \wedge \neg d) \vee (o \wedge \neg d)$ ,  $Q_{K_2} = (\neg s \wedge d \wedge \neg o) \vee (\neg s \wedge \neg d \wedge o)$ , and  $Q_{K_3} = s \wedge d \wedge o$ . As we can see in the fifth column of Table 1 the models of  $PS\text{-Merge}(E)$  are the states  $(0, 0, 1)$  and  $(1, 0, 1)$ .

In [8] two classes of merging operators are defined: majority and arbitration merging. The former strives to satisfy a maximum of agents' beliefs and the latter tries to satisfy each agent's beliefs to the best possible degree. The former notion is treated in the context of  $PS\text{-Merge}$ , and it can be refined tending to arbitration if we calculate the minimum value among the Partial Satisfiability

<sup>1</sup> If  $\Delta$  is a merging operator, we are going to abuse the notation by referring to the models of the merging operator  $mod(\Delta(E))$  and their respective belief base  $\Delta(E)$  simply as  $\Delta(E)$ .



**Table 1.** *PS-Merge* of Example 3 and min function

$w$	$Q_{K_1}$	$Q_{K_2}$	$Q_{K_3}$	$Sum$	$min$
(1, 1, 1)	$\frac{1}{2}$	$\frac{1}{3}$	1	$\frac{11}{6} \approx 1.83$	$\frac{1}{3}$
(1, 1, 0)	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{11}{6} \approx 1.83$	$\frac{1}{3}$
(1, 0, 1)	1	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{14}{6} \approx \mathbf{2.33}$	$\frac{1}{3}$
(1, 0, 0)	1	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{10}{6} \approx 1.67$	$\frac{1}{3}$
(0, 1, 1)	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{11}{6} \approx 1.83$	$\frac{1}{3}$
(0, 1, 0)	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{9}{6} = 1.5$	$\frac{1}{3}$
(0, 0, 1)	1	1	$\frac{1}{2}$	$\frac{14}{6} \approx \mathbf{2.33}$	$\frac{1}{3}$
(0, 0, 0)	$\frac{1}{2}$	$\frac{1}{3}$	0	$\frac{7}{6} \approx 1.67$	0

of the bases. Then with this indicator, we have a form to choose the state that is impartial and tries to satisfy all agents as far as possible. If we again consider Example 3 in Table 1 there are two different states that maximize the sum of the Partial Satisfaction of the profile, (1, 0, 1) and (0, 0, 1). If we try to minimize the individual dissatisfaction these two states do not provide the same results. Using the *min* function (see 6<sup>th</sup> column of Table 1) over the partial satisfaction of the bases we get the states that minimize the individual dissatisfaction and between the states (1, 0, 1) and (0, 0, 1) obtained by the proposal we might prefer the state (1, 0, 1) over (0, 0, 1) as the  $\Delta_{GMax}$  operator (an arbitration operator) does in 3.

It is possible to extend this notion of *PS-Merge* in the case where a set of integrity constraints must be obeyed. If  $\mu$  is a formula representing the set of integrity constraints, then the states that falsify the integrity constraint cannot be considered in the *PS-Merge*. If  $\mathcal{W}(\mu)$  denotes the set of states that validate the integrity constraints, it is enough to restrict the definition of the Partial Satisfiability Merge to  $\mathcal{W}(\mu)$ .

**Definition 3.** Let  $E$  be a belief profile obtained from the belief bases  $K_1, \dots, K_m$ . Then  $PS-Merge_\mu(E)$ , the Partial Satisfiability Merge of  $E$  given the set of integrity constraints  $\mu$ , is a mapping from the belief profiles to belief bases such that the set of models of the resulting base is:

$$\left\{ w \in \mathcal{W}(\mu) \mid \sum_{i=1}^m w_{ps}(Q_{K_i}) \geq \sum_{i=1}^m w'_{ps}(Q_{K_i}) \text{ for all } w' \in \mathcal{W}(\mu) \right\}$$

*Example 4.* (See 8) At a meeting of four co-owners of a block of flats, the chairman proposes the construction of a swimming-pool, a tennis-court and a private-car-park in the coming year. But if two of these three items are built, the rent will increase significantly. We will denote by  $s$ ,  $t$  and  $p$  the construction of the swimming-pool, the tennis-court and the private car-park respectively and  $i$  will denote the increase of the rent. Two co-owners want to build the three items, and do not care about the rent increase ( $K_1 = K_2 = s \wedge t \wedge p$ ), the third thinks that building any item will cause at some time an increase of the rent and wants to pay the lowest rent so he is opposed to any construction (so

$K_3 = \neg s \wedge \neg t \wedge \neg p \wedge \neg i$ ) and finally the last one thinks that the flat really needs a tennis-court and a private car-park but does not want a rent increase (i.e.  $K_4 = t \wedge p \wedge \neg i$ ).

The chairman outlines that building two or more items will increase the rent significantly. This fact cannot be ignored and the states in which this fact is falsified must be ignored. These kind of facts are known as integrity constraints. In our example the integrity constraints  $\mu$  are represented by the single formula  $((s \wedge t) \vee (s \wedge p) \vee (t \wedge p)) \rightarrow i$ . If we consider  $P$  the ordered set  $\{s, t, p, i\}$  then the states  $(1, 1, 1, 0)$ ,  $(1, 1, 0, 0)$ ,  $(1, 0, 1, 0)$  and  $(0, 1, 1, 0)$  cannot be considered as a possible Partial Satisfiability Merge since these states falsify the integrity constraint. It is enough to calculate the Partial-Satisfiability to states in  $\mathcal{W}(\mu)$ .

The answer to example 4 obtained by applying *PS-Merge* is the state  $(1, 1, 1, 1)$ , i.e. the decision that satisfies the majority of the group is to build the three items no matter if the rent increases. This decision is also the one obtained using the integrity constraint majority merging operator based in the  $\Sigma$  operator in [8].

## 4 Properties

Finding a set of axiomatic properties that an operator may satisfy in order to exhibit a rational behavior is a concern greatly studied. In [3, 4, 2, 5] sets of postulates have been proposed concerning belief merging operators.

In [3, 11] Konieczny and Pino-Pérez proposed the basic properties (A1)-(A6) for merging operators, rephrased without reference to integrity constraints.

**Definition 4.** *Let  $E, E_1, E_2$  be belief profiles, and  $K_1$  and  $K_2$  be consistent belief bases. Let  $\Delta$  be an operator which assigns to each belief profile  $E$  a belief base  $\Delta(E)$ .  $\Delta$  is a merging operator if and only if it satisfies the following postulates:*

(A1)  $\Delta(E)$  is consistent

(A2) if  $\bigwedge E$  is consistent then  $\Delta(E) \equiv \bigwedge E$

(A3) if  $E_1 \equiv E_2$ , then  $\Delta(E_1) \equiv \Delta(E_2)$

(A4)  $\Delta(\{K_1, K_2\}) \wedge K_1$  is consistent if and only if  $\Delta(\{K_1, K_2\}) \wedge K_2$  is consistent

(A5)  $\Delta(E_1) \wedge \Delta(E_2) \models \Delta(E_1 \sqcup E_2)$

(A6) if  $\Delta(E_1) \wedge \Delta(E_2)$  is consistent, then  $\Delta(E_1 \sqcup E_2) \models \Delta(E_1) \wedge \Delta(E_2)$

The intuitive meaning of the postulates is as follows: (A1) ensures the extraction of a piece of information from the profile. (A2) states that if the belief bases agree on some alternatives, then the result of the merging will be these alternatives. (A3) ensures that the operator obeys a principle of irrelevance of syntax. (A4) is the fairness postulate, such that when we merge two bases the operator should not give preference to one of them. (A5) expresses the following: if we have two groups viewed as profiles  $E_1$  and  $E_2$ , and  $E_1$  compromises a set of alternatives to which  $A$  belongs, and  $E_2$  compromises another set which also

contains  $A$ , then if we join the two groups  $A$  must be in the chosen alternatives. (A5) and (A6) together state that if one could find two groups which agree on at least one alternative, then the result of the global merging will be exactly these alternatives.

We analyze the minimal set of properties *PS-Merge* satisfies and its rational behavior concerning merging. Clearly *PS-Merge* satisfies (A1), which simply requires of the result of merging to be consistent. *PS-Merge* also satisfies (A2).

**Proposition 1.**  $\bigwedge E \not\perp \perp$  implies  $PS\text{-Merge}(E) \equiv \bigwedge E$ .

*Proof.* Let  $E = \{Q_{K_1}, \dots, Q_{K_m}\}$  be a profile with its beliefs bases expressed in DNF such that  $\bigwedge E \not\perp \perp$ . There are  $l > 0$  states  $w_1, \dots, w_l$  that satisfy each base thus for every state  $w_r$  we can find  $m$  disjoints  $d_1, \dots, d_m$  belonging to each base  $Q_{K_1}, \dots, Q_{K_m}$  respectively, that are satisfied by  $w_r$ . Consequently the Partial Satisfiability of the bases for every  $w_r$  is evaluated in 1, i.e.  $w_{rps}(Q_{K_j}) = 1$  for  $1 \leq r \leq l$  and  $1 \leq j \leq m$ . So we can affirm that  $\sum_{i=1}^m w_{rps}(Q_{K_i}) = m$  for each model of the profile. Notice that every disjoint can have either of two values  $\sum_{i=1}^s \frac{w(C_i)}{s}$  or  $\frac{n - |\bigwedge_{i=1}^s C_i|}{2n}$  (see Definition [II](#)). Moreover the first value is less or equal to 1 and the second one is less or equal to  $\frac{n}{2n} = \frac{1}{2}$ . From this fact we can affirm that if a state  $w$  does not satisfy a base  $Q_K$  then  $w_{ps}(Q_K) < 1$  and we can conclude that  $\sum_{i=1}^m w_{ps}(Q_{K_i}) < m$  for the states that do not satisfy the profile. Hence a state  $w$  is included in the merge iff  $w$  is a model of  $\bigwedge E$ , i.e. we obtain only models of the conjunction of the bases as a result of *PS-Merge* when the profile is consistent.  $\square$

The next property (A3) is a version of Dalal’s principle of the Irrelevance of Syntax [\[12\]](#). In general, *PS-Merge* does not satisfy (A3). Consider the situation, called implicit knowledge in [\[13\]](#), where systems want to extract additional knowledge that is not locally held by any agent. For example, if an agent knows  $a$  and another agent knows  $a \rightarrow b$ , then combining their knowledge yields  $b$ , whereas neither one of them individually knows it. Using most of the merging operators we can find the expected result. Now suppose that this situation is presented in the mind of an agent, i.e. both facts  $a$  and  $a \rightarrow b$  are known by an agent who does not know how to combine the facts in order to produce  $b$  and hence its beliefs in DNF are  $K_1 = (a \wedge \neg a) \vee (a \wedge b)$ . On the other hand suppose another agent who knows explicitly that  $a$  and  $b$  hold, i.e. its beliefs in DNF are  $K_2 = a \wedge b$ . We can see that both agents’ bases are equivalent. Now using *PS-Merge* to combine the bases with another agent’s base  $K_3 = \neg b$ , we obtain the states (1, 0) and (0, 0) from merging  $K_1$  and  $K_3$  and only the state (1, 0) from merging  $K_2$  and  $K_3$ . *PS-Merge* is a majority operator which tries to satisfy each base as much as possible. Hence in the first case the maximum percentage of satisfaction for  $K_1$  is 50% if it wants to leave a percentage of satisfaction for  $K_3$  different from 0%, noticing that state (1, 0) satisfies  $a$  and (0, 0) satisfies  $a \rightarrow b$ . In the second case where  $K_2$  is satisfied 50% by (1, 0), we can see that if the agent knows explicitly the facts then *PS-Merge* refines the answer. We can also see that even though (A3) is not satisfied by *PS-Merge*, the results show a realistic behavior. The result of combining information without make inferences

beforehand might not be as preferable as when agents find some consequences of their knowledge before the merging.

In general, *PS-Merge* does not satisfy (A4). Consider again  $K_1 = (a \wedge \neg a) \vee (a \wedge b)$  and  $K_3 = \neg b$ . We can see that both bases are consistent by themselves however their conjunction is not. As we know from the example above, using *PS-Merge* to combine them we obtain the states (1, 0) and (0, 0) which clearly prefer  $K_3$ .  $K_1$  shows an indecision of the agent  $\neg a \vee b$  which is why the merging process prefers the satisfaction of the “confident” source. However if *PS-Merge* takes as parameters bases showing only explicit information, for example  $K_2 = a \wedge b$  and  $K_3 = \neg b$ , the merging process does not lead to a preference for any of them. The result of the example is state (1, 0) which is not the model of either of them.

If there is not “redundant” information, i.e. formulas including disjoints of the style  $a \wedge \neg a$ , (A3) and (A4) are satisfied. *PS-Merge* satisfies the property (A4) under certain restrictions.

**Proposition 2.**  $\Delta(\{K_1, K_2\}) \wedge K_1 \not\models \perp$  iff  $\Delta(\{K_1, K_2\}) \wedge K_2 \not\models \perp$ .

(A5) and (A6) establish connections between two results; the result obtained when merging each of two belief profiles and then taking their conjunction and the result obtained when first combining the two belief profiles and then performing a single merge. Together the two properties require that these two results be equivalent, provided that the conjunction referenced is not inconsistent. *PS-Merge* satisfies (A5) but it is necessary to consider that profiles come from different contexts and they can have different languages. In this case it will be necessary to extend the language of  $E_1$  to include the atoms appearing in  $E_2$  and vice versa.

**Proposition 3.** If  $\mathcal{P}(E_1) = \mathcal{P}(E_2)$  then  $PS-Merge(E_1) \wedge PS-Merge(E_2) \models PS-Merge(E_1 \sqcup E_2)$

*Proof.* If  $PS-Merge(E_1) \wedge PS-Merge(E_2)$  is consistent then each model  $w$  of the conjunction maximizes the Partial Satisfaction of  $E_1$  and  $E_2$  at the same time because  $w$  is model of each merging. I.e.  $\sum_{k_i \in E_1} w_{ps}(Q_{K_i}) \geq \sum_{k_i \in E_1} w'_{ps}(Q_{K_i})$  and  $\sum_{k_i \in E_2} w_{ps}(Q_{K_i}) \geq \sum_{k_i \in E_2} w'_{ps}(Q_{K_i})$  for all  $w' \in \mathcal{W}$ . The merging of the union of the profiles is simply the sum of the Partial Satisfaction of the profiles  $E_1$  and  $E_2$ . Then for all  $w' \in \mathcal{W}$ :

$$\begin{aligned} \sum_{k_i \in E_1 \sqcup E_2} w_{ps}(Q_{K_i}) &= \sum_{k_i \in E_1} w_{ps}(Q_{K_i}) + \sum_{k_i \in E_2} w_{ps}(Q_{K_i}) \geq \\ &\sum_{k_i \in E_1} w'_{ps}(Q_{K_i}) + \sum_{k_i \in E_2} w'_{ps}(Q_{K_i}) = \sum_{k_i \in E_1 \sqcup E_2} w'_{ps}(Q_{K_i}) \end{aligned}$$

□

*Remark 1.* By definition the *PS-Merge* is commutative. I.e. the result of the merging does not depend on any order of the bases of the profile.

As stated before there are two important classes of merging operators, majority and arbitration operators. The behavior of majority operators is to say that if an opinion is the most popular, then it will be the opinion of the group. A postulate that captures this idea is the postulate (M7) of [3].

$$(M7) \forall K \exists n \in \mathbb{N} \quad \Delta(E \sqcup \{K\}^n) \models K$$

*PS-Merge* satisfies the postulate (M7) as a direct consequence of the definition of *PS-Merge*. Even more, the definition of *PS-Merge* not only tries to satisfy the majority of the group, it also tries to satisfy to the maximum degree (see example 10 in the following section). *PS-Merge* does not satisfy all the postulates (A1)-(A6), however, it behaves as a majority merging operator. As the reader can see in the next section the behavior of *PS-Merge* is close to  $\Delta_{\Sigma}$  and *CMerge* which are majority operators.

### 5 Comparing Results

*PS-Merge* yields similar results compared with existing techniques such as *CMerge*, the  $\Delta_{\Sigma}$ <sup>2</sup> operator and *MCS* (Maximal Consistent Subsets) considered in [5][3][8]. Let  $E$  be in each case the belief profile consisting of the belief bases enlisted below and let  $P$  be corresponding set of atoms ordered alphabetically.

1.  $K_1 = K_2 = \{a\}$  and  $K_3 = \{\neg a\}$ .  $CMerge(E) = \{a\}$  which is equivalent to  $PS-Merge(E) = \Delta_{\Sigma}(E) = \{(1)\}$ .
2.  $K_1 = \{b\}$ ,  $K_2 = \{a, a \rightarrow b\}$  and  $K_3 = \{\neg b\}$ . In this case  $CMerge(E) = \{a, a \rightarrow b\}$  which is equivalent to  $PS-Merge(E) = \Delta_{\Sigma}(E) = \{(1, 1)\}$ .
3.  $K_1 = \{b\}$ ,  $K_2 = \{a, b\}$  and  $K_3 = \{\neg b\}$ .  $CMerge(E)$  and the model obtained from  $\Delta_{\Sigma}$  and  $PS-Merge(E)$  are as in the previous case.
4.  $K_1 = \{b\}$ ,  $K_2 = K_3 = \{a \rightarrow b\}$  and  $K_4 = \{a, \neg b\}$ .  $CMerge(E) = \{a, a \rightarrow b\}$  and  $PS-Merge(E) = \Delta_{\Sigma}(E) = \{(1, 1)\}$  which are all equivalent.
5.  $K_1 = \{a, c\}$ ,  $K_2 = \{a \rightarrow b, \neg c\}$  and  $K_3 = \{b \rightarrow d, c\}$ . Here  $CMerge(E) = \{a, a \rightarrow b, b \rightarrow d, c\}$  which is equivalent to  $PS-Merge(E) = \Delta_{\Sigma}(E) = \{(1, 1, 1, 1)\}$ .
6.  $K_1 = \{a, c\}$ ,  $K_2 = \{a \rightarrow b, \neg c\}$ ,  $K_3 = \{b \rightarrow d, c\}$  and  $K_4 = \{\neg c\}$ . While  $CMerge(E) = MCS(E) = \{a, a \rightarrow b, b \rightarrow d\}$  which is equivalent to  $\Delta_{\Sigma}(E) = \{(1, 1, 0, 1), (1, 1, 1, 1)\}$ ,  $PS-Merge(E) = \{(1, 1, 0, 1)\}$ . *CMerge*, *MCS* and the  $\Delta_{\Sigma}$  operator give no information about  $c$ . Using *PS-Merge*,  $c$  is falsified and this leads us to have total satisfaction of the second and fourth bases and partial satisfaction of the first and third bases.
7.  $K_1 = \{a\}$ ,  $K_2 = \{a \rightarrow b\}$  and  $K_3 = \{a, \neg b\}$ . Now  $CMerge(E) = \{a\}$ ,  $\Delta_{\Sigma}(E) = \{(1, 1), (1, 0)\}$  and  $PS-Merge(E) = \{(1, 1)\}$ . The model  $(1, 0)$  satisfies only two bases while the model  $(1, 1)$  satisfy two bases and a “half” of the third base.

---

<sup>2</sup> As stated in [3], merging operator  $\Delta_{\Sigma}$  is equivalent to the merging operator proposed by Lin and Mendelzon in [5] called *CMerge*.

8.  $K_1 = \{a\}$ ,  $K_2 = \{a \rightarrow b\}$ ,  $K_3 = \{a, \neg b\}$  and  $K_4 = \{\neg b\}$ .  $CMerge(E) = \{a \wedge \neg b\}$ , which is equivalent to  $PS-Merge(E) = \Delta_\Sigma(E) = \{(1, 0)\}$ .
9.  $K_1 = \{b\}$ ,  $K_2 = \{a \rightarrow b\}$  and  $K_3 = \{a, \neg b\}$ . Now  $CMerge(E) = \{a \wedge b\}$  and  $PS-Merge(E) = \Delta_\Sigma(E) = \{(1, 1)\}$ .
10.  $K_1 = \{b\}$ ,  $K_2 = \{a \rightarrow b\}$ ,  $K_3 = \{a, \neg b\}$  and  $K_4 = \{\neg b\}$ .  $CMerge(E) = \{a \vee \neg b\}$ ,  $\Delta_\Sigma(E) = \{(0, 0), (1, 0), (1, 1)\}$  and  $PS-Merge(E) = \{(1, 1), (0, 0)\}$ . The model  $(1, 0)$  obtained using  $\Delta_\Sigma$  operator satisfies only two bases, while the two options of  $PS-Merge(E)$  satisfy two bases and a “half” of the third base. Then  $PS-Merge$  is a refinement of the answer given by  $CMerge$  and  $\Delta_\Sigma$ .
11.  $K_1 = K_2 = \{a \wedge b \wedge c\}$ ,  $K_3 = \{\neg a \wedge \neg b \wedge \neg c \wedge \neg d\}$  and  $K_4 = \{b \wedge c \wedge \neg d\}$  with the restriction that if two of  $a$ ,  $b$  or  $c$  are validated it forces  $d$  to be validated as well.  $CMerge(E) = \{a \wedge b \wedge c \wedge d\}$ ,  $PS-Merge = \Delta_\Sigma(E) = \{(1, 1, 1, 1)\}$ .

## 6 Conclusion

A merging operator has been proposed in [7] that is not defined in terms of a distance measure, but is Partial Satisfiability-based. It appears to resolve conflicts among the beliefs bases in a natural way. The idea is intended to extend the notion of satisfiability to one that includes a “measure” of satisfaction. This notion of satisfaction considers that whenever an atom does not appear in a formula then it is considered that the agent has no preferences on this literal so a partial satisfaction different from 0 is assigned. The reader can see from Definition 1 that  $\frac{1}{2}$  is chosen. This measure considers the intuitive idea that an “or” is satisfied if any of its disjuncts is satisfied and in the case of an “and” we count the number of conjuncts satisfied. We can think that a state always satisfies a formula by a percentage, which is given by the Partial Satisfiability. Once a satisfaction measure of belief bases is given, it is used to define  $PS-Merge$ . Unlike the operators proposed in the literature, in order to know the “degree” of satisfaction by a given state,  $PS-Merge$  does not need to calculate a partial pre-order over the set of states since Partial Satisfiability can be calculated for a single state. In this way the comparison between states becomes easier. However it is necessary to take into account that before calculating the Partial Satisfiability of a formula it is necessary to transform it into DNF.

Unlike other approaches  $PS-Merge$  can consider beliefs bases which are inconsistent, since the source of inconsistency can refer to specific atoms and the operator takes into account the rest of the information.

The approach bears some resemblance to the belief merging framework proposed in [3,8,5,9], particularly with the  $\Delta_\Sigma$  operator. As with those approaches the  $Sum$  function is used, but instead of using it to measure the distance between the states and the profile  $PS-Merge$  uses  $Sum$  to calculate the general degree of satisfiability. The result of  $PS-Merge$  are simply the states which maximize the  $Sum$  of the Partial Satisfiability of the profile and it is not necessary to define a partial pre-order. Because of this similarity between  $PS-Merge$  and  $\Delta_\Sigma$  we propose to analyze this similarity in term of the postulates satisfied by  $\Delta_\Sigma$  outlined in [3,8]. In this paper we analysed of some of the postulates, and even

thought the *PS-Merge* does not satisfy all the properties cited in [3,5] it has a rational behavior.

As in [8] in order to consider integrity constraints *PS-Merge* selects the states among the states which validate the integrity constraints rather than in  $\mathcal{W}$ . The approach behaves as a majority operator but an arbitration operator can also be defined in terms of Partial Satisfiability in a similar way.

As future work a further analysis of the *PS-Merge* is necessary to characterize its behaviour in terms of postulates. As well, study of the properties of the approach in cases where integrity constraints are almost parallel to the simple case is required. It remains for the definition of an arbitration operator in terms of Partial Satisfiability and the corresponding characterization to be considered. Finally it is necessary to study the complexity of the whole process of the *PS-Merge* in order to compare it with the existing techniques.

## References

1. Bloch, I., Hunter, A.: Fusion: General concepts and characteristics. *International Journal of Intelligent Systems* 10(16), 1107–1134 (2001)
2. Liberatore, P., Schaerf, M.: Arbitration (or how to merge knowledge bases). *IEEE Transactions on Knowledge and Data Engineering* 10(1), 76–90 (1998)
3. Konieczny, S., Pino-Pérez, R.: On the logic of merging. In: Cohn, A.G., Schubert, L., Shapiro, S.C. (eds.) *Principles of Knowledge Representation and Reasoning*, pp. 488–498. Morgan Kaufmann, San Francisco (1998)
4. Revesz, P.Z.: On the Semantics of Arbitration. *Journal of Algebra and Computation* 7 (2), 133–160 (1997)
5. Lin, J., Mendelzon, A.: Knowledge base merging by majority. In: Pareschi, R., B, F. (eds.) *Dynamic Worlds: From the Frame Problem to Knowledge Management*. Kluwer Academic Publishers, Dordrecht (1999)
6. Baral, C., Kraus, S., Minker, J., Subrahmanian, V.: Combining knowledge bases consisting of first-order theories. *Computational Intelligence* 1(8), 45–71 (1992)
7. Borja Macías, V., Pozos Parra, P.: Model-based belief merging without distance measures. In: *International Conference on Autonomous Agents and Multiagent Systems*, Honolulu, Hawaii, pp. 613–615. ACM Press, New York (2007)
8. Konieczny, S., Pino-Pérez, R.: Merging with integrity constraints. In: Hunter, A., Parsons, S. (eds.) *ECSQARU 1999*. LNCS (LNAI), vol. 1638, Springer, Heidelberg (1999)
9. Meyer, T., Pozos, P., Perrussel, L.: Mediation using m-states. In: Godo, L. (ed.) *ECSQARU 2005*. LNCS (LNAI), vol. 3571, pp. 489–500. Springer, Heidelberg (2005)
10. Revesz, P.Z.: On the semantics of theory change: Arbitration between old and new information. In: *Proceedings of the Twelfth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 71–82. ACM Press, New York (1993)
11. Konieczny, S., Pino-Pérez, R.: Merging information under constraints: a logical framework. *Journal of Logic and Computation* 12 (5), 773–808 (2002)
12. Dalal, M.: Investigations into a theory of knowledge base revision. In: *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence*, Saint Paul, Minnesota, pp. 475–479 (1988)
13. Halpern, J., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence* 3(54), 319–379 (1992)



# Optimizing Inference in Bayesian Networks and Semiring Valuation Algebras

Michael Wachter<sup>1</sup>, Rolf Haenni<sup>2</sup>, and Marc Pouly<sup>3</sup>

<sup>1</sup> University of Bern, Switzerland  
{wachter, haenni}@iam.unibe.ch

<sup>2</sup> Bern University of Applied Sciences, Switzerland  
rolf.haenni@bfh.ch

<sup>3</sup> University of Fribourg, Switzerland  
marc.pouly@unifr.ch

**Abstract.** Previous work on context-specific independence in Bayesian networks is driven by a common goal, namely to represent the conditional probability tables in a most compact way. In this paper, we argue from the view point of the knowledge compilation map and conclude that the language of Ordered Binary Decision Diagrams (OBDD) is the most suitable one for representing probability tables, in addition to the language of Algebraic Decision Diagrams (ADD). We thus suggest the replacement of the current practice of using tree-based or rule-based representations. This holds not only for inference in Bayesian networks, but is more generally applicable in the generic framework of semiring valuation algebras, which can be applied to solve a variety of inference and optimization problems in different domains.

## 1 Introduction

*Bayesian networks* (BN) are a very flexible and powerful tool in many areas, particularly in AI related problems and applications [1]. Its power stems from the efficient encoding of independence relations among variables. Originally, BNs were mainly designed to exploit so-called *conditional* (or *structural*) *independences*, which allows the (global) joint probability function to be replaced by several (local) *conditional probability tables* (CPT). The locality of the CPTs in turn is responsible for the success of BNs as an efficient computational tool for probabilistic inference [2].

The exploitation of another type of independence relations, so-called *contextual* or *context-specific independences* (CSI), has been proposed in [3]. CSI deals with local independence relations *within* (rather than *between*) the given CPTs. A *context* within a CPT is a partial parent configuration.

---

<sup>1</sup> The notion of context-specific independence first appeared in the influence diagram literature [4]. Note that some authors prefer to use *contextual strong independence* as an alternative name with the same acronym [5]. Other similar notions are *asymmetric independence* [6] and *probabilistic causal irrelevance* [7].



Most approaches to exploit CSI suggest a tree-structured CPT representation, but different names such as *CPT-trees* [3], *probability trees* [8], or *multi-resolution binary trees* [9] are in use for essentially the same concept. All these techniques share a common goal, namely to merge CPT entries with the same value for a specific context. Note that such a simplified CPT may still include the same value more than once.

More advanced CPT representations allow a complete partitioning of the parent configurations, in which each value occurs exactly once. A simple idea to achieve this is to represent the partitions by logical rules [10], but a more efficient approach is the use of *Algebraic Decision Diagrams* (ADD) as suggested in [11]. Note that ADDs are a generalization of *Ordered Binary Decision Diagrams* (OBDD) [12]. Technically speaking, this method exceeds CSI insofar as it considers the entire local structure to simplify a given CPT, thus possibly spanning over various contexts.

In this paper, we start by looking at the exploitation of local CPT structures from the perspective of the *knowledge compilation map* [13,14]. This map supports the identification of the most appropriate representation language according to the *queries* and *transformations* it is supposed to offer in polynomial time. At the end, the main conclusion from this view will be the following:

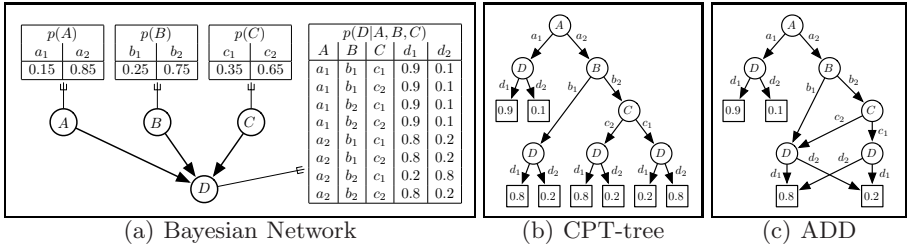
*The languages of OBDDs (and possibly DNFs) is the most appropriate representation language for local CPT structures in BNs, in addition to the language of ADDs.*

To obtain this result and to emphasize its generality, we will entirely shift our analysis from Bayesian networks into the generic framework of *semiring valuation algebras* [15,16]. This is an abstract theory of inference in knowledge-based systems, which is based on two principal operations called *combination* and *variable elimination* (or *marginalization*). The generality of valuation algebras allows us to extend the applicability of the above results from BNs to a much broader range of formalisms and applications thereof. Moreover, it opens up new opportunities and possibilities for building generic approximations methods.

The structure of this paper is as follows. Section 2 provides a short summary of inference in BNs. Section 3 is devoted to semiring valuations and their connection to BNs. Section 4 discusses the optimization of the underlying representation. Section 5 concludes the paper.

## 2 Inference in Bayesian Networks

A *Bayesian network* (BN) is an efficient representation of a *joint probability mass function* over a set  $\mathbf{X}$  of variables [1]. We assume throughout this paper that all variables  $X \in \mathbf{X}$  are *binary*, i.e. their associated sets of possible values are  $\Omega_X = \{x_1, x_2\}$ . The network itself consists of a directed acyclic graph (DAG), which represents the direct influences among the variables, each of them attached to one node, and a set of conditional probability tables (CPT), which quantify the strengths of these influences. The whole BN represents a joint probability mass function  $p : \Omega_{\mathbf{X}} \rightarrow [0, 1]$  over its variables in a compact manner by



**Fig. 1.** Example of a simple Bayesian network with four variables, and two other representations of the CPT  $p(D|A, B, C)$

$$p(\mathbf{X}) = \prod_{X \in \mathbf{X}} p(X|\text{parents}(X)), \tag{1}$$

where  $\text{parents}(X)$  denotes the parents of node  $X$  in the DAG. Figure 1(a) depicts a simple BN. It consists of four variables  $A, B, C,$  and  $D,$  with corresponding CPTs for  $p(A), p(B), p(C),$  and  $p(D|A, B, C).$

Inference in Bayesian networks means to compute the conditional probability  $P(H=h | E_1=e_1, \dots, E_r=e_r),$  or simply

$$P(h|\mathbf{e}) = \frac{P(h, \mathbf{e})}{P(\mathbf{e})}, \tag{2}$$

of a hypothesis  $h \in \Omega_H$  for some observed evidence  $\mathbf{e} = (e_1, \dots, e_r) \in \Omega_{\mathbf{E}}.$  We will call the elements of  $\mathbf{E} = \{E_1, \dots, E_r\} \subseteq \mathbf{X}$  *evidence variables.* To see how to solve the inference problem, let  $\mathbf{Y} = \{Y_1, \dots, Y_s\} \subseteq \mathbf{X}$  be an arbitrary subset of variables,  $\mathbf{y} = (y_1, \dots, y_s) \in \Omega_{\mathbf{Y}}$  a configuration of values  $y_i \in Y_i,$  and  $\mathbf{Z} = \mathbf{X} \setminus \mathbf{Y}.$  Then it is sufficient to compute

$$P(\mathbf{y}) = \sum_{\mathbf{z} \in \Omega_{\mathbf{Z}}} p(\mathbf{y}\mathbf{z}) \tag{3}$$

twice, once with  $\mathbf{Y} = \{H\} \cup \mathbf{E}$  and  $\mathbf{y} = (h, \mathbf{e})$  to get the nominator and once with  $\mathbf{Y} = \mathbf{E}$  and  $\mathbf{y} = \mathbf{e}$  to get the denominator of the above formula. Note that the necessary sum-of-products involve exponentially many terms relative to  $|\mathbf{Z}|,$  but if the computations are performed *locally* in a join tree propagation or variable elimination process, it is almost always possible to replace it by a very compact factorization [2][7][18]. Local computation is a generic inference technique for all sorts of valuation algebras (see Section 3).

In the presence of context-specific independence, further efficiency improvements are possible. Consider the following simplified version of the original definition.

**Definition 1 (Boutilier et al., 1996).** *If  $\mathbf{X}, \mathbf{Y},$  and  $\mathbf{Z}$  are pairwise disjoint sets of variables, then  $\mathbf{X}$  is context-specific independent of  $\mathbf{Y}$  in the context  $\mathbf{z} \in \Omega_{\mathbf{Z}},$  if  $p(\mathbf{X}|\mathbf{Y}, \mathbf{z}) = p(\mathbf{X}|\mathbf{z})$  whenever  $p(\mathbf{Y}, \mathbf{z}) > 0.$*

In the example of Figure 1(a),  $\{D\}$  is context-specific independent of  $\{B, C\}$  in the context  $a_1$ . Similarly,  $\{D\}$  is context-specific independent of  $\{C\}$  in the context  $(a_2, b_1)$ , and so on.

The classical approach to exploit context-specific independence is to use tree-based CPT representations [3,8,9]. An example of such a *CPT-tree* (or *probability tree*) is depicted in Figure 1(b) for the CPT  $p(D|A, B, C)$  in the BN of Figure 1(a). Each node in the tree represents a decision w.r.t. the possible values of the indicated variable, and the values attached to the terminal nodes are the conditional probabilities  $p(d_i|\mathbf{z})$ , where  $\mathbf{z}$  denotes the context specified by the path up to the root.

A more sophisticated CPT representation has been proposed in [11] to speed up the logical compilation of BNs. The idea is to use (ordered) *algebraic decision diagrams* [19], an extension of OBDDs to multiple terminal nodes. In the particular application of representing CSI, one can simply think of an ADD as a CPT-tree in which all identical nodes are merged, as shown in Figure 1(c). The result is obviously a more compact CPT representations. In extreme cases, ADDs are even exponentially smaller than corresponding CPT-trees. Nevertheless, ADDs inherit all the nice computational properties from OBDDs.

### 3 Inference in Valuation Algebras

To enlarge the applicability of the above ideas, the analysis is now shifted from BNs into the generic theory of *valuation algebras* [15]. The theory's basic elements are *valuations*, which can be regarded as pieces of information about the possible values of some variables. Thus, if  $\mathbf{X}$  denotes the set of all variables relevant to a problem, then each valuation  $\varphi$  refers to a finite set of variables  $d(\varphi) \subseteq \mathbf{X}$ , called its *domain*. For an arbitrary set  $\mathbf{Y} \subseteq \mathbf{X}$  of variables,  $\Phi_{\mathbf{Y}}$  denotes the set of all valuations  $\varphi$  with  $d(\varphi) = \mathbf{Y}$ . With this notation, we can write

$$\Phi = \bigcup_{\mathbf{Y} \subseteq \mathbf{X}} \Phi_{\mathbf{Y}} \tag{4}$$

to denote the set of all possible valuations over  $\mathbf{X}$ . If  $2^{\mathbf{X}}$  denotes the powerset<sup>2</sup> of  $\mathbf{X}$ , then

- *Labeling*:  $\Phi \rightarrow 2^{\mathbf{X}}, \varphi \mapsto d(\varphi)$ ;
- *Combination*:  $\Phi \times \Phi \rightarrow \Phi, (\varphi, \psi) \mapsto \varphi \otimes \psi$ ;
- *Variable elimination*:  $\Phi \times \mathbf{X} \rightarrow \Phi, (\varphi, X) \mapsto \varphi^{-X}$ ;

are the three primitive operations of a valuation algebra.

**Definition 2 (Kohlas, 2003).** *A tuple  $(\Phi, 2^{\mathbf{X}}, d, \otimes, -)$  is a valuation algebra, if it satisfies the following set of axioms:*

<sup>2</sup> The more general definition given in [15] considers arbitrary distributive lattices. In this case, we must replace the operation of variable elimination by marginalization.

1. Commutative Semigroup:  $\Phi$  is associative and commutative under  $\otimes$ .
2. Labeling: If  $\varphi, \psi \in \Phi$ , then  $d(\varphi \otimes \psi) = d(\varphi) \cup d(\psi)$ .
3. Variable Elimination: If  $\varphi \in \Phi$  and  $X \in d(\varphi)$ , then  $d(\varphi^{-X}) = d(\varphi) - \{X\}$ .
4. Commutativity of Elimination: If  $\varphi \in \Phi_{\mathbf{X}}$  and  $X, Y \in d(\varphi)$ , then  $(\varphi^{-X})^{-Y} = (\varphi^{-Y})^{-X}$ .
5. Combination: If  $\varphi, \psi \in \Phi$  with  $X \notin d(\varphi)$  and  $X \in d(\psi)$ , then  $(\varphi \otimes \psi)^{-X} = \varphi \otimes \psi^{-X}$ .

Instances of valuation algebras are large in number and occur in very different contexts. One of the most prominent instances are the CPTs of BNs, where multiplication and summation over tables are the operations of combination and variable elimination, respectively. Valuations of this particular type are often called *probability potentials* [20]. For an extensive list of valuation algebra instances, we refer to [15].

### 3.1 Semiring Valuations

An important class of valuation algebras, which actually covers a majority of the known instances, results from the notion of *semiring valuations*. For this, let the elements  $X \in \mathbf{X}$  be binary variables with frames  $\Omega_X = \{x_1, x_2\}$ .<sup>3</sup> If  $\mathbf{Y} \subseteq \mathbf{X}$  is a subset of variables, then the Boolean vectors  $\mathbf{y} \in \Omega_{\mathbf{Y}}$  are called configurations of  $\mathbf{Y}$ . By convention, we define the frame of the empty variable set as  $\Omega_{\emptyset} = \{\diamond\}$ . Furthermore, we write  $\mathbf{y}^{\downarrow \mathbf{Z}}$  for the projection of some configuration  $\mathbf{y} \in \Omega_{\mathbf{Y}}$  to a subset  $\mathbf{Z} \subseteq \mathbf{Y}$ . In particular, we have  $\mathbf{y}^{\downarrow \emptyset} = \diamond$ .

Consider now a (commutative) *semiring*  $\mathcal{A} = \langle A, +, \times \rangle$ , i.e. an algebraic structure over a set of values  $A$ , where the operations  $+$  and  $\times$  are both associative and commutative, and where  $\times$  distributes over  $+$ .

**Definition 3 (Kohlas, 2004).** A semiring valuation  $\varphi$  with domain  $d(\varphi) = \mathbf{Y}$  is a mapping  $\varphi : \Omega_{\mathbf{Y}} \rightarrow A$  from the set of configurations  $\Omega_{\mathbf{Y}}$  to the set of values  $A$  of a semiring  $\mathcal{A} = \langle A, +, \times \rangle$ .

With respect to the set  $\Phi$  of all semiring valuations over the variables  $\mathbf{X}$ , the operations of combination and variable elimination are defined in terms of the semiring operations  $+$  and  $\times$ :

- *Combination*: for  $\mathbf{Y}, \mathbf{Z} \subseteq \mathbf{X}$ ,  $\varphi \in \Phi_{\mathbf{Y}}$ ,  $\psi \in \Phi_{\mathbf{Z}}$ , and  $\mathbf{x} \in \Omega_{\mathbf{Y} \cup \mathbf{Z}}$ , let

$$\varphi \otimes \psi(\mathbf{x}) := \varphi(\mathbf{x}^{\downarrow d(\varphi)}) \times \psi(\mathbf{x}^{\downarrow d(\psi)}).$$

- *Variable Elimination*: for  $\mathbf{Y} \subseteq \mathbf{X}$ ,  $\varphi \in \Phi_{\mathbf{Y}}$ ,  $X \in \mathbf{Y}$ , and  $\mathbf{z} \in \Omega_{\mathbf{Y} \setminus \{X\}}$ , let

$$\varphi^{-X}(\mathbf{z}) := \varphi(\mathbf{z}, x_1) + \varphi(\mathbf{z}, x_2).$$

---

<sup>3</sup> The theory of semiring valuation algebras can be developed with arbitrary finite variables. Here, we restrict ourselves to binary variables which will allow us later to identify configurations with models of Boolean functions. Note that this is no conceptual restriction [21].

$\varphi_A$	$a_1$	$a_2$
	0.15	0.85

$\varphi_B$	$b_1$	$b_2$
	0.25	0.75

$\varphi_C$	$c_1$	$c_2$
	0.35	0.65

$\varphi_{D A,B,C}$	$a_1$	$a_1$	$a_1$	$a_1$	$a_1$	$a_1$	$a_1$	$a_1$	$a_2$	$a_2$	$a_2$	$a_2$	$a_2$	$a_2$	$a_2$	$a_2$
	$b_1$	$b_1$	$b_1$	$b_1$	$b_2$	$b_2$	$b_2$	$b_2$	$b_1$	$b_1$	$b_1$	$b_1$	$b_2$	$b_2$	$b_2$	$b_2$
	$c_1$	$c_1$	$c_2$	$c_2$	$c_1$	$c_1$	$c_2$	$c_2$	$c_1$	$c_1$	$c_2$	$c_2$	$c_1$	$c_1$	$c_2$	$c_2$
	$d_1$	$d_2$	$d_1$	$d_2$	$d_1$	$d_2$	$d_1$	$d_2$	$d_1$	$d_2$	$d_1$	$d_2$	$d_1$	$d_2$	$d_1$	$d_2$
	0.9	0.1	0.9	0.1	0.9	0.1	0.9	0.1	0.8	0.2	0.8	0.2	0.8	0.2	0.8	0.8

**Fig. 2.** The semiring valuations  $\varphi_A$ ,  $\varphi_B$ ,  $\varphi_C$ , and  $\varphi_{D|A,B,C}$  representing the CPTs  $p(A)$ ,  $p(B)$ ,  $p(C)$ , and  $p(D|A, B, C)$  of the BN in Figure 1(a)

The most important property of semiring valuations is described in the following theorem.

**Theorem 1 (Kohlas, 2004; Kohlas & Wilson, 2006).** *A set  $\Phi$  of semiring valuations, with labeling, combination and variable elimination as defined above, satisfies the axioms of a valuation algebra.*

The insight that every semiring induces a valuation algebra foreshadows the richness of formalisms that are covered by this theory. If we take for example the *arithmetic semiring*  $(\mathbb{R}_0^+, +, *)$ , we obtain the valuation algebra of probability potentials, which are commonly used as a CPT representation for BNs [20]. Figure 2 illustrates this for the CPTs of the BN in Figure 1(a).

### 3.2 Local Computation

The computational interest in valuation algebras arises from the following notion of an *inference problem* and the generality of the resulting solution. For a given set of valuations  $\{\varphi_1, \dots, \varphi_n\}$ , called the *knowledge base*, the inference problem consists in eliminating from the joint valuation  $\varphi = \varphi_1 \otimes \dots \otimes \varphi_n$  with  $d(\varphi) = \mathbf{X}$  all variables that do not belong to some set  $\mathbf{Q} \subseteq \mathbf{X}$  of *query variables*. More formally, this means the computation of

$$\varphi^{-\mathbf{X} \setminus \mathbf{Q}} = (\varphi_1 \otimes \dots \otimes \varphi_n)^{-\mathbf{X} \setminus \mathbf{Q}}. \tag{5}$$

Note that the transitivity of the variable elimination allows us to eliminate sets of variables without further specifying the ordering (see Axiom 4).

To solve the inference problem efficiently, it is clear that an explicit computation of the joint valuation is normally not feasible. Local computation methods counteract this problem by organizing the computations in such a way that the maximal domain size remains reasonably bounded. In the following, we restrict our attention to one such algorithm called *fusion algorithm* [23] and refer to [24] for a broad discussion of related local computation schemes.

<sup>4</sup> In most cases, the complexity of valuation algebra operations tends to increase exponentially with the size of the involved domains.

<sup>5</sup> Other names for exactly the same type of algorithm are *bucket elimination* [18] or simply *variable elimination* [22].

To describe the fusion algorithm, we consider first the elimination of a single variable  $X$  from a set of valuations  $\Psi \subseteq \Phi$ , which is defined by

$$\text{Fus}_X(\Psi) := \{\psi_X^{-X}\} \cup \{\varphi \in \Psi : X \notin d(\varphi)\} \quad (6)$$

with  $\psi_X = \otimes\{\varphi \in \Psi : X \in d(\varphi)\}$ . The fusion algorithm follows then from a repeated application of this basic operation to all variables in  $\mathbf{X} \setminus \mathbf{Q} = \{X_1, \dots, X_k\}$ . This leads to the following general solution for the inference problem:

$$\begin{aligned} \varphi^{-\mathbf{X} \setminus \mathbf{Q}} &= (\varphi_1 \otimes \dots \otimes \varphi_n)^{-\{X_1, \dots, X_k\}} \\ &= \otimes \text{Fus}_{X_k}(\dots (\text{Fus}_{X_1}(\{\varphi_1, \dots, \varphi_n\}) \dots)). \end{aligned} \quad (7)$$

We refer to [15] for a proof and further considerations regarding the complexity of this generic inference algorithm.

In the particular case of probabilistic inference in a BN, the fusion algorithm has to be performed twice, once for the query variables  $\mathbf{H} = \{H\} \cup \mathbf{E}$  and once for  $\mathbf{Q} = \mathbf{E}$  (see Section 2). The resulting valuation  $\varphi^{-\mathbf{X} \setminus \mathbf{H}}$  contains the probabilities  $P(h, \mathbf{e})$  of all configurations  $(h, \mathbf{e}) \in \Omega_{\mathbf{H}}$ . Similarly,  $\varphi^{-\mathbf{X} \setminus \mathbf{Q}}$  contains the probabilities  $P(\mathbf{e})$  of all configurations  $\mathbf{e} \in \Omega_{\mathbf{Q}} = \Omega_{\mathbf{E}}$ . This means that  $P(h|\mathbf{e}) = P(h, \mathbf{e})/P(\mathbf{e})$  can be derived from the corresponding semiring values of  $(h, \mathbf{e})$  in  $\varphi^{-\mathbf{X} \setminus \mathbf{H}}$  and  $\mathbf{e}$  in  $\varphi^{-\mathbf{X} \setminus \mathbf{Q}}$ , respectively.<sup>6</sup>

## 4 Compact Representations

An optimized CPT representation is important to further speed up inference in BNs, i.e. to go beyond the capacities offered by local computation. The CSI approach with its tree-structured representations (see Section 2) is a good starting point, but now we will show that we can do better than that. For this, we will no longer look at identical CPT entries as the result of CSI, but instead consider them from a purely technical point of view within the generic framework of semiring valuations. This will then allow us to use the knowledge compilation map to select the most appropriate representation language.

### 4.1 Partitioned Semiring Valuations

The basic idea consists in partitioning the configurations space  $\Omega_{\mathbf{Y}}$  of a semiring valuation  $\varphi$  with  $d(\varphi) = \mathbf{Y}$  according to its semiring values into a collection  $\{S_1, \dots, S_s\}$  of exclusive and exhaustive subsets  $S_i \subseteq \Omega_{\mathbf{Y}}$ . In other words, instead of mapping single configurations into (possibly identical) semiring values, we will now map partitions of configurations  $S$  into (pairwise distinct) semiring values  $\varphi(S) \in A$ . A valuation represented in this way will be called *partitioned semiring valuation*. Note that in the extreme case, where  $\varphi(\mathbf{y}) = c$  is the same

<sup>6</sup> This way of using the fusion algorithm may not always be optimal, especially if  $\mathbf{E}$  is large. A better idea is to create for each evidence variable  $E \in \mathbf{E}$  an additional *evidence valuation*  $\varphi_E$  with  $d(\varphi_E) = \{E\}$  and  $\varphi_E(e) = 1$ , and to apply the fusion algorithm for  $\mathbf{H} = \{H\}$  and  $\mathbf{Q} = \emptyset$  to the extended set of valuations.

$\varphi_{D A,B,C}$		
$S_1$	$\{(a_1, b_1, c_1, d_1), (a_1, b_1, c_2, d_1), (a_1, b_2, c_1, d_1), (a_1, b_2, c_2, d_1)\}$	0.9
$S_2$	$\{(a_1, b_1, c_1, d_2), (a_1, b_1, c_2, d_2), (a_1, b_2, c_1, d_2), (a_1, b_2, c_2, d_2)\}$	0.1
$S_3$	$\{(a_2, b_1, c_1, d_1), (a_2, b_1, c_2, d_1), (a_2, b_2, c_1, d_2), (a_2, b_2, c_2, d_1)\}$	0.8
$S_4$	$\{(a_2, b_1, c_1, d_2), (a_2, b_1, c_2, d_2), (a_2, b_2, c_1, d_1), (a_2, b_2, c_2, d_2)\}$	0.2

(a) Sets

$f_i : \Omega_{\{A,B,C,D\}} \rightarrow \{0, 1\}$		
$f_1$	$a_1 \wedge d_1$	0.9
$f_2$	$a_1 \wedge d_2$	0.1
$f_3$	$a_2 \wedge ((b_1 \wedge d_1) \vee (b_2 \wedge ((c_1 \wedge d_2) \vee (c_2 \wedge d_1))))$	0.8
$f_4$	$a_2 \wedge ((b_1 \wedge d_2) \vee (b_2 \wedge ((c_1 \wedge d_1) \vee (c_2 \wedge d_2))))$	0.2

(b) Boolean Functions

**Fig. 3.** The partitioning of the configurations in the semiring valuation  $\varphi_{D|A,B,C}$  into sets of configurations and their representation as Boolean functions

constant value  $c \in A$  for all  $\mathbf{y} \in \Omega_{\mathbf{Y}}$ , we will end up with a single partition  $\Omega_{\mathbf{Y}}$  with  $\varphi(\Omega_{\mathbf{Y}}) = c$ . Figure 3(a) shows the partitioned semiring valuation  $\varphi_{D|A,B,C}$  of Figure 2.

With this idea in mind, it is clear that the question of representing semiring valuations becomes a question of representing sets of configurations, i.e. subsets of a Cartesian product. In the case of binary variables, we can identify such a set  $S \subseteq \Omega_{\mathbf{Y}}$  with a *Boolean function* (BF)  $f : \Omega_{\mathbf{Y}} \rightarrow \{0, 1\}$ , which evaluates to 1 for all  $\mathbf{y} \in S$  and to 0 for all  $\mathbf{y} \notin S$ . Then  $S$  becomes the so-called *satisfying set* of  $f$ . With this, our problem of representing semiring valuations turns into a problem of representing Boolean functions<sup>7</sup>.

The optimal representation of a BF is a lively research topic with contributions from many different areas. To get a good survey of the vast number of existing techniques and their relationships, the most convenient and comprehensive access is the *knowledge compilation map* in [13] and its extension in [14]. The goal of this map is to support the identification of the most appropriate representation language according to the *queries* and *transformations* it is supposed to offer in polynomial time. Therefore, as soon as we know which queries and transformations are required for dealing with partitioned semiring valuations, we can use the map to identify the most appropriate language.

In the following, we use  $\mathbf{0}$  to denote the constant BF that always evaluates to 0. Similarly,  $\mathbf{1}$  denotes the constant BF that always evaluates to 1. Furthermore,  $x_i \in \Omega_X$  represents the BF which evaluates to 1 iff  $X = x_i$ . If  $f_1$  and  $f_2$  are Boolean functions, then  $f_1 \wedge f_2$  is the BF that evaluates to 1, iff both  $f_1$  and  $f_2$  evaluate to 1. Similarly,  $f_1 \vee f_2$  denotes the BF that evaluates to 1, iff either

<sup>7</sup> The more general case of non-binary variables leads to general indicator functions, for which similar representation languages and an analogue knowledge compilation map exist [21].

$f_1$  or  $f_2$  evaluates to 1. Figure 3(b) shows the partitioned semiring valuation  $\varphi_{D|A,B,C}$  in terms of their BFs.

## 4.2 Queries and Transformations

To determine the required queries and transformations, the two essential operations of combination and variable elimination have to be analyzed in the light of the suggested representation. The labeling operation is negligible, since it can be achieved easily. In the following, let  $(f_i, v_i)$ ,  $i \in \{1, \dots, s\}$ , be the entries of a partitioned semiring valuation  $\varphi \in \Phi_{\mathbf{Y}}$ . With  $f_i$  we denote the BF of the partition  $S_i$  and with  $v_i = \varphi(S_i)$  the corresponding semiring value. Similarly, let  $(g_j, w_j)$ ,  $j \in \{1, \dots, t\}$ , be the entries of another partitioned semiring valuation  $\psi \in \Phi_{\mathbf{Z}}$ , where  $g_j$  denotes the BF of the partition  $T_j$  and  $w_j = \psi(T_j)$  the corresponding semiring value.

Let us now discuss the combination and variable elimination according to the definitions given in Section 3.

**Combination.** It is quite obvious that  $\varphi \otimes \psi$  essentially consists of all combined entries  $(f_i \wedge g_j, v_i \times w_j)$ , except of the ones with  $f_i \wedge g_j = \mathbf{0}$ . In the terminology of [14],  $f_i \wedge g_j$  corresponds to the transformation “*binary conjunction*”, denoted by  $\text{AND}_2$ , and the test  $f_i \wedge g_j = \mathbf{0}$  corresponds to the query “*consistency test*”, denoted by  $\text{CO}$ . Hence, both  $\text{AND}_2$  and  $\text{CO}$  are required for the combination of two partitioned semiring valuations.

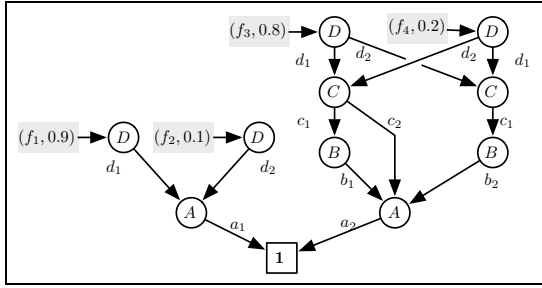
**Variable Elimination.** For the elimination of a variable  $X \in d(\varphi)$  from  $\varphi$ , we will now see that  $\varphi^{-X}$  can be expressed in terms of an operation similar to the above combination. For this, let  $\varphi|x$  denote the result of “*conditioning*”  $\varphi$  on a value  $x \in \Omega_X$  using the method proposed in [13]. The idea is to delete from each partition  $S_i$  the configurations which do not contain  $x$ , and then project all remaining configurations to  $\mathbf{Y} \setminus X$ . In terms of the involved BFs, this can be achieved by conditioning each  $f_i$  on  $x$ . Note that  $f_i|x$  may become equivalent to  $\mathbf{0}$  for some  $i \in \{1, \dots, s\}$ . In [14], the conditioning of a Boolean function is denoted by  $\text{TC}$  (for *term conditioning*).

In the binary case, i.e. for  $\Omega_X = \{x_1, x_2\}$ , let  $I, J \subseteq \{1, \dots, s\}$  contain the indices of each  $f_i|x_1 \neq \mathbf{0}$  resp.  $f_i|x_2 \neq \mathbf{0}$ . With this, we obtain  $\varphi^{-X}$  by computing all combined entries  $(f_i|x_1 \wedge f_j|x_2, v_i + v_j)$  for  $i \in I$  and  $j \in J$ . Note that we may again get and delete some functions equivalent to  $\mathbf{0}$ . Eliminating a variable from a partitioned semiring valuations requires thus  $\text{TC}$ ,  $\text{AND}_2$ , and  $\text{CO}$ .

To complete our analysis of the above operations, consider the case where some of the resulting partitions obtain the same semiring value  $a \in A$ . This may occur frequently<sup>8</sup> (especially if  $A$  is small) and for both the combination (as a result of the semiring operation  $\times$ ) and the variable elimination (as a result of the semiring operation  $+$ ). To merge two such entries  $(f_1, a)$  and  $(f_2, a)$  into  $(f_1 \vee f_2, a)$ , an

<sup>8</sup> Or never, such as in the compilation method described in [11], where  $A$  itself is a set of Boolean functions.





**Fig. 4.** The OBDDs representing the BF’s of the semiring valuation  $\varphi_{D|A,B,C}$ . Edges leading towards 0 and the node 0 itself are omitted.

additional transformation  $OR_2$  (called *binary disjunction*) is required. In general, we may need to merge several entries  $(f_1, a), \dots, (f_k, a)$  into  $(f_1 \vee \dots \vee f_k, a)$ , which requires the transformation  $OR$  (called *general disjunction*).

### 4.3 Selecting the Optimal Representation Language

Now that we know that working with partitioned semiring valuations requires  $TC$ ,  $CO$ ,  $AND_2$ , and  $OR_2$  (or preferably  $OR$ ), we may use the knowledge compilation map in [13,14] to select the most appropriate language. It turns out that two languages are valuable candidates:

- OBDDs with a fixed variable ordering support  $TC$ ,  $CO$ ,  $AND_2$ , and  $OR_2$  (but not  $OR$ ) in polynomial time;
- DNFs support  $TC$ ,  $CO$ ,  $AND_2$ ,  $OR_2$ , and  $OR$  in polynomial time.

No other language offers  $CO$  and  $AND_2$  together. Note that in terms of their succinctness, the OBDD and DNF languages are incomparable [13]. Nevertheless, OBDDs are often much smaller than corresponding DNFs, which is why we recommend OBDDs to be used as representation language for partitioned semiring valuations<sup>9</sup>. Note that the proposed rule-based representation proposed in [10] is very close to using DNFs.

Figure 4 shows the OBDDs representing the BF’s  $f_1$ ,  $f_2$ ,  $f_3$ , and  $f_4$  of the partitioned semiring valuation  $\varphi_{D|A,B,C}$  together with the associated semiring values. Note that there is some substantial overlap between the OBDDs of  $f_3$  and  $f_4$ . Such overlapping OBDD structures are called *shared OBDDs* in [25].

## 5 Conclusion

This paper has two main messages. The first one is that the results in the context of optimizing the representation of a BN can be applied to semiring valuations. We also conclude that OBDDs (and possibly DNFs) form the most appropriate language. Our argumentation is entirely based on the knowledge compilation

<sup>9</sup> In the knowledge compilation map, the DNF language is situated in the family of *flat* languages [13], which are generally not very attractive.

map and covers the special case of representing CPTs (and CSI) in Bayesian networks. Future work will focus on elaborating a generic approximation method, its implementation, and experimental evaluation thereof. In addition, we will analyze the relationship between shared OBDD and ADD representations, which appear to be closely connected.

## Acknowledgements

This research is supported by the *Swiss National Science Foundation*, Project No. PP002-102652/1 & Grant No. 200020-109510, and *The Leverhulme Trust*.

## References

1. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Mateo (1988)
2. Shenoy, P.P., Shafer, G.: Axioms for probability and belief-function propagation. In: Shachter, R.D., Levitt, T.S., Lemmer, J.F., Kanal, L.N. (eds.) UAI 1988. 4th Conference on Uncertainty in Artificial Intelligence, Minneapolis, USA, pp. 169–198 (1988)
3. Boutilier, C., Friedman, N., Goldszmidt, M., Koller, D.: Context-specific independence in Bayesian networks. In: Horvitz, E., Jensen, F. (eds.) UAI 1996. 12th Conference on Uncertainty in Artificial Intelligence, Portland, USA, pp. 115–123 (1996)
4. Smith, J.E., Holtzman, S., Matheson, J.E.: Structuring conditional relationships in influence diagrams. *Operations Research* 41(2), 280–297 (1993)
5. Wong, S.K., Butz, C.: Contextual weak independence in Bayesian networks. In: Laskey, K.B., Prade, H. (eds.) UAI 1999. 15th Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, pp. 670–679 (1999)
6. Geiger, D., Heckerman, D.: Advances in probabilistic reasoning. In: UAI 1991. 6th Annual Conference on Uncertainty in Artificial Intelligence, Los Angeles, USA, pp. 118–126 (1991)
7. Galles, D., Pearl, J.: Axioms of causal relevance. *Artificial Intelligence* 97(1–2), 9–43 (1997)
8. Cano, A., Moral, S., Salmeron, A.: Penniless propagation in join trees. *International Journal of Intelligent Systems* 15(11), 1027–1059 (2000)
9. Bellot, D., Bessière, P.: Approximate discrete probability distribution representation using a multi-resolution binary tree. In: ICTAI 2003. 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento, USA, pp. 498–503 (2003)
10. Poole, D., Zhang, N.L.: Exploiting contextual independence in probabilistic inference. *Journal of Artificial Intelligence Research* 18, 263–313 (2003)
11. Chavira, M., Darwiche, A.: Compiling Bayesian networks using variable elimination. In: IJCAI 2007. 20th International Joint Conference on Artificial Intelligence, Hyderabad, India (2007)
12. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers* 35(8), 677–691 (1986)
13. Darwiche, A., Marquis, P.: A knowledge compilation map. *Journal of Artificial Intelligence Research* 17, 229–264 (2002)

14. Wachter, M., Haenni, R.: Propositional DAGs: a new graph-based language for representing Boolean functions. In: Doherty, P., Mylopoulos, J., Welty, C. (eds.) KR 2006. 10th International Conference on Principles of Knowledge Representation and Reasoning, pp. 277–285. AAAI Press, USA (2006)
15. Kohlas, J.: Information Algebras: Generic Structures for Inference. Springer, London (2003)
16. Kohlas, J., Wilson, N.: Exact and approximate local computation in semiring induced valuation algebras. Technical Report 06–06, Department of Informatics, University of Fribourg, Switzerland (2006)
17. Kohlas, J., Shenoy, P.P.: Computation in valuation algebras. In: Gabbay, D.M., Smets, P. (eds.) Handbook of Defeasible Reasoning and Uncertainty Management Systems. Algorithms for Uncertainty and Defeasible Reasoning, vol. 5, pp. 5–39. Kluwer Academic Publishers, Dordrecht (2000)
18. Dechter, R.: Bucket elimination: a unifying framework for reasoning. *Artificial Intelligence* 113(1–2), 41–85 (1999)
19. Bahar, R.I., Frohm, E.A., Gaona, C.M., Hachtel, G.D., Macii, E., Pardo, A., Somenzi, F.: Algebraic decision diagrams and their applications. In: Lightner, M.R., Jess, J.A.G. (eds.) 4th IEEE/ACM International Conference on Computer-Aided Design, pp. 188–191. IEEE Computer Society Press, California (1993)
20. Shenoy, P.P., Shafer, G.: Axioms for probability and belief function propagation. In: Shafer, G., Pearl, J. (eds.) *Readings in Uncertain Reasoning*, pp. 575–610. Morgan Kaufmann, USA (1990)
21. Wachter, M., Haenni, R.: Multi-state directed acyclic graphs. In: Kobti, Z., Wu, D. (eds.) CanAI 2007. 20th Canadian Conference on Artificial Intelligence, Montréal, Canada. LNCS (LNAI), vol. 4509, pp. 464–475 (2007)
22. Zhang, N.L., Poole, D.: Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research* 5, 301–328 (1996)
23. Shenoy, P.P.: Valuation-based systems: A framework for managing uncertainty in expert systems. In: Zadeh, L.A., Kacprzyk, J. (eds.) *Fuzzy Logic for the Management of Uncertainty*, pp. 83–104. John Wiley and Sons, New York (1992)
24. Schneuwly, C., Pouly, M., Kohlas, J.: Local computation in covering join trees. Technical Report 04–16, Department of Informatics, University of Fribourg, Switzerland (2004)
25. Wegener, I.: *Branching Programs and Binary Decision Diagrams – Theory and Applications*. Number 56 in *Monographs on Discrete Mathematics and Applications*. SIAM (2000)

# Compiling Solution Configurations in Semiring Valuation Systems

Marc Pouly<sup>1</sup>, Rolf Haenni<sup>2,3</sup>, and Michael Wachter<sup>3</sup>

<sup>1</sup>University of Fribourg, Switzerland

marc.pouly@unifr.ch

<sup>2</sup>Bern University of Applied Sciences, Switzerland

rolf.haenni@bfh.ch

<sup>3</sup>University of Bern, Switzerland

{wachter, haenni}@iam.unibe.ch

**Abstract.** This paper describes a new method for solving optimization queries in semiring valuation systems. In contrast to existing techniques which focus essentially on the identification of solution configurations, we propose foremost the construction of an implicit representation of the solution configuration set in the shape of a Boolean function. This intermediate compilation step allows then to efficiently execute many further relevant queries that go far beyond the traditional task of enumerating solution configurations.

## 1 Introduction

A valuation algebra [1,2,3] is an abstract system that unifies various formalisms of knowledge representation which are all based on the two principal operations of combination and variable elimination. Based on this generic framework, a collection of efficient inference algorithms was developed which deal successfully with the exponential nature behind the operations. These computational architectures are referred to as local computation algorithms. Thereupon, [4] remarked that the general inference problem turns into an optimization task when dealing with valuation algebras whose variable elimination operator corresponds in some way to either minimization or maximization. Consequently, an extended local computation scheme was proposed that allows furthermore to identify the configurations that map to the computed optimum value. Under this perspective, local computation methods turn into *non-serial dynamic programming* [5]. However, practical applications in diagnosis often make further demands that go far beyond the identification of solution configurations. Possible scenarios range from the simple requirement of counting solution configurations without their explicit enumeration to the question about the probability of single configurations or the equivalence of objective functions. To allow for these requirements, we propose the construction of a Boolean function whose models collapse with the solution configurations we are looking for. By choosing then a suitable representation of the latter, we obtain a very compact graphical structure that makes the fast evaluation of such queries possible. Because this compilation process is by itself based on local computation, we inherit also its efficiency. With this proposition, we follow recent ambitions [6,7] in combining local computation with knowledge compilation techniques [8,9].

This paper is organized in the following way. We first give a quick introduction to valuation algebras, the generic inference problem and the fusion algorithm for its efficient solution. Next, we introduce a particular class of valuation algebras that map configurations to semiring values. On the according semiring, we can identify sufficient conditions such that the inference problem turns into an optimization task. This rounds off the needed background and allows us to introduce this new technique of solving optimization tasks in semiring valuation systems.

## 2 Valuation Algebras

The basic elements of a valuation algebra are so-called *valuations*. Intuitively, a valuation can be regarded as a representation of knowledge about the possible values of a set of variables  $r$ . It can be said that each valuation  $\phi$  refers to a finite set of variables  $d(\phi) \subseteq r$ , called its *domain*. For an arbitrary set  $s \subseteq r$  of variables,  $\Phi_s$  denotes the set of valuations  $\phi$  with  $d(\phi) = s$ . With this notation, the set of all possible valuations over  $r$  can be defined as

$$\Phi = \bigcup_{s \subseteq r} \Phi_s.$$

Let  $2^r$  be the power set<sup>1</sup> of  $r$  and  $\Phi$  a set of valuations with their domains in  $2^r$ . We assume the following operations defined in  $(\Phi, 2^r)$ :

- *Labeling*:  $\Phi \rightarrow 2^r$ ;  $\phi \mapsto d(\phi)$ ,
- *Combination*:  $\Phi \times \Phi \rightarrow \Phi$ ;  $(\phi, \psi) \mapsto \phi \otimes \psi$ ,
- *Variable Elimination*:  $\Phi \times r \rightarrow \Phi$ ;  $(\phi, X) \mapsto \phi^{-X}$ .

These are the three basic operations of a valuation algebra. We impose now the following set of axioms:

1. *Commutative Semigroup*:  $\Phi$  is associative and commutative under  $\otimes$ .
2. *Labeling*: For  $\phi, \psi \in \Phi$ ,  $d(\phi \otimes \psi) = d(\phi) \cup d(\psi)$ .
3. *Variable Elimination*: For  $\phi \in \Phi$  and  $X \in d(\phi)$ ,

$$d(\phi^{-X}) = d(\phi) - \{X\}.$$

4. *Commutativity of Elimination*: For  $\phi \in \Phi_s$  and  $X, Y \in s$ ,

$$(\phi^{-X})^{-Y} = (\phi^{-Y})^{-X}.$$

5. *Combination*: For  $\phi, \psi \in \Phi$  with  $X \notin d(\phi)$ ,  $X \in d(\psi)$ ,

$$(\phi \otimes \psi)^{-X} = \phi \otimes \psi^{-X}.$$

Instances of valuation algebras are large in number and occur in very different contexts. Among them are probability mass functions used in Bayesian networks, Dempster-Shafer belief functions, relations from database theory or more general constraint systems, various kinds of logics and many more. We refer to [3] for an extensive listing of instances.

<sup>1</sup> The more general definition given in [3] considers arbitrary distributive lattices. In this case, we must replace the operation of variable elimination by marginalization.

### 3 Inference Problem and Local Computation

The computational interest in valuation algebras is generally resumed as *inference problem*. Given a set of valuations  $\{\phi_1, \dots, \phi_n\}$ , we need to eliminate all variables from the joint valuation  $\phi = \phi_1 \otimes \dots \otimes \phi_n$  that do not belong to some query  $x \subseteq r$ . More formally, this consists in the computation of

$$\phi^{-\{X_1, \dots, X_k\}} = (\phi_1 \otimes \dots \otimes \phi_n)^{-\{X_1, \dots, X_k\}} \quad (1)$$

for  $\{X_1, \dots, X_k\} = d(\phi) - x$ . Note that we are allowed to eliminate sets of variables because their elimination order is not decisive (Axiom 4). It is however well-known that an explicit computation of the joint valuation is out of question because in most cases, the complexity of valuation algebra operations tends to increase exponentially with the size of the involved factor domains. Local computation methods counteract this problem by organizing the computations in such a way that factors do not grow significantly [3]. In the following, we restrict our attention to one such algorithm called *fusion algorithm* [10] and refer to [11] for a broad discussion of further local computation schemes.

In order to describe the fusion algorithm, we consider first the elimination of a single variable  $Y$  from a set of valuations. This operation can be performed as follows:

$$\text{Fus}_Y(\{\phi_1, \dots, \phi_n\}) = \{\psi^{-Y}\} \cup \{\phi_i : Y \notin d(\phi_i)\},$$

where

$$\psi = \bigotimes_{i: Y \in d(\phi_i)} \phi_i.$$

The fusion algorithm follows then by a repeated application of this operation:

$$\phi^{-\{X_1, \dots, X_k\}} = \bigotimes \text{Fus}_{X_k}(\dots (\text{Fus}_{X_1}(\{\phi_1, \dots, \phi_n\})).$$

We refer to [3] for a proof and further considerations regarding the complexity of this generic inference algorithm.

### 4 Semirings

Semirings are algebraic structures with two binary operations  $+$  and  $\times$  over a set of values  $A$ . We call a tuple  $\mathcal{A} = \langle A, +, \times \rangle$  a *semiring* if both operations  $+$  and  $\times$  are associative and commutative and if  $\times$  distributes over  $+$ . Elsewhere, this is also called a *commutative semiring*. If there is an element  $\mathbf{0} \in A$  such that  $\mathbf{0} + a = a + \mathbf{0} = a$  and  $\mathbf{0} \times a = a \times \mathbf{0} = \mathbf{0}$  for all  $a \in A$ , then  $\mathcal{A}$  is called a semiring with *zero element*. A zero element can be adjoined to any semiring such that we can assume its existence without loss of generality. If addition is idempotent, i.e. if  $a + a = a$  for all  $a \in A$ , we call  $\mathcal{A}$  an *idempotent semiring*. Finally, a semiring element  $\mathbf{1} \in A$  is said to be a *unit element* if  $\mathbf{1} \times a = a \times \mathbf{1} = a$  for all  $a \in A$ . We can extend any idempotent semiring to include a unit element and therefore, we assume subsequently that all idempotent semirings possess a unit element. The following table lists some of the most famous semiring examples which will later be reconsidered in the discussion of semiring valuation algebras.

	<b>A</b>	<b>+</b>	<b>×</b>	<b>0</b>	<b>1</b>	<b>idempotent</b>
1	$\mathbb{R}_{\geq 0}$	+	$\cdot$	0	1	$\times$
2	$\mathbb{R} \cup \{\pm\infty\}$	max	min	$-\infty$	$\infty$	$\checkmark$
3	$\mathbb{B} = \{0, 1\}$	$\vee$	$\wedge$	0	1	$\checkmark$
4	$\mathbb{N} \cup \{0, \infty\}$	min	+	$\infty$	0	$\checkmark$
5	$\mathbb{R} \cup \{-\infty\}$	max	+	$-\infty$	0	$\checkmark$
6	$[0, 1]$	max	t-norm	0	1	$\checkmark$
7	$\{f : \mathbb{B}^v \rightarrow \mathbb{B}\}$	max	min	$f_0$	$f_1$	$\checkmark$

The first semiring is called *Arithmetic Semiring* and consists of the non-negative reals together with the usual operations of addition and multiplication. The second example is named *Bottleneck Semiring* and often used in the context of optimization tasks. Next follows the famous *Boolean Semiring* with disjunction and conjunction for addition and multiplication respectively. Examples 4 and 5 are known as *Tropical Semirings* and Example 6 shows that maximization together with any t-norm induces again a semiring. T-norms are binary operations on the unit interval which are commutative, associative, have the number 1 as unit element and are non-decreasing in both arguments [12]. Finally, Example 7 shows the semiring of Boolean functions over a set of propositional variables  $v$ . Here,  $f_0$  ( $f_1$ ) denotes the constant Boolean function that always maps to 0 (1).

### 4.1 Ordered Semirings

On every idempotent semiring  $\mathcal{A}$ , we can introduce a relation  $\leq_{id}$  by:

$$a \leq_{id} b \text{ if and only if } a + b = b.$$

This relation is a partial order and both semiring operations  $+$  and  $\times$  are monotone with respect to it. Furthermore, we have for all elements  $a, b \in A$  that  $a + b = \sup\{a, b\}$ . However, in many interesting semirings the relation  $\leq_{id}$  is actually a total order which leads to the following refinement of this result:

**Lemma 1.** *If  $\leq_{id}$  is total, we have  $a + b = \max\{a, b\}$ .*

Finally, we require another property to guarantee that all solution configurations are captured during the compilation process. Indeed, it was shown in [13] that only a non-empty subset of the solution configurations are found in case of its absence. Such an example is given by the Bottleneck semiring in the above table.

**Definition 1.** *An idempotent semiring is called strictly monotonic over  $\times$  if for  $c \neq 0$ ,  $a <_{id} b$  implies that  $a \times c <_{id} b \times c$ .*

## 5 Semiring Valuation Algebras

Let  $r$  be a set of propositional variables<sup>2</sup>. Without abandoning the usual interpretation of frame elements as truth values, we denote the frame of variable  $X$  as  $\Omega_X = \{0_X, 1_X\}$ .

<sup>2</sup> The theory of semiring induced valuation algebras can be developed with arbitrary finite variables. Here, we restrict ourselves to propositional variables for simplicity reasons.

Correspondingly, the frame  $\Omega_s$  of a non-empty variable set  $s \subseteq r$  is built by the Cartesian product

$$\Omega_s = \prod_{X \in s} \Omega_X.$$

The Boolean vectors  $\mathbf{x} \in \Omega_s$  are called configurations of  $s$ , and by convention, we define the frame of the empty variable set as  $\Omega_\emptyset = \{\diamond\}$ . Furthermore, we write  $\mathbf{x}^{\downarrow t}$  for the projection of some configuration  $\mathbf{x} \in \Omega_s$  to a subset  $t$  of  $s$ . In particular, we have  $\mathbf{x}^{\downarrow \emptyset} = \diamond$ .

A *semiring valuation*  $\phi$  with domain  $s \subseteq r$  is defined to be a function that associates a value from a given semiring  $\mathcal{A} = \langle A, +, \times \rangle$  with each configuration  $\mathbf{x} \in \Omega_s$ , i.e.

$$\phi : \Omega_s \rightarrow A.$$

Again, we refer to the set of all semiring valuations over variables in  $s$  as  $\Phi_s$  and define  $\Phi$  to be their union over all subsets  $s \subseteq r$ . Thus, the operation of labeling for semiring valuations can be defined as follows:

1. *Labeling*:  $d(\phi) = s$  if  $\phi \in \Phi_s$ .

Next, we define the operations of combination and variable elimination for semiring valuations in terms of the semiring operations  $+$  and  $\times$ :

2. *Combination*:  $\phi \in \Phi_s, \psi \in \Phi_t$  and  $\mathbf{x} \in \Omega_{s \cup t}$

$$\phi \otimes \psi(\mathbf{x}) = \phi(\mathbf{x}^{\downarrow d(\phi)}) \times \psi(\mathbf{x}^{\downarrow d(\psi)}). \tag{2}$$

3. *Variable Elimination*:  $\phi \in \Phi_s, X \in s$  and  $\mathbf{x} \in \Omega_{s - \{X\}}$

$$\phi^{-X}(\mathbf{x}) = \phi(\mathbf{x}, 0_X) + \phi(\mathbf{x}, 1_X). \tag{3}$$

**Theorem 1.** *A system of semiring valuations with labeling, combination and variable elimination as defined above, satisfies the axioms of a valuation algebra.*

A proof of this important theorem can be found in [14]. The insight that every semiring induces a valuation algebra foreshadows the richness of formalisms that are covered by this theory. If we take for example the Arithmetic Semiring restricted to the unit interval, we obtain the valuation algebra of probability potentials [1] which represent the conditional probability functions in Bayesian networks. In the same way, the Boolean semiring induces the valuation algebra of indicator functions that is used in various constraint-based applications [15] and, as another example, the t-norm semiring leads to the valuation algebra of possibility potentials [16].

## 6 Optimization Problems

The inference problem, as stated in Equation (1), adopts a very special meaning in the case of valuation algebras that are induced by totally ordered idempotent semirings. According to Lemma 1 we obtain by elimination of a subset  $t \subseteq d(\phi) = s$  of variables

$$\phi^{-t}(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega_t} \phi(\mathbf{x}, \mathbf{y}) = \max\{\phi(\mathbf{x}, \mathbf{y}), \mathbf{y} \in \Omega_t\}.$$



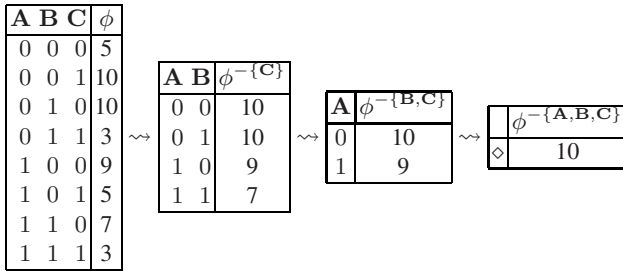
In particular, we obtain after eliminating all variables in  $s$

$$\phi^{-s}(\diamond) = \max\{\phi(\mathbf{x}), \mathbf{x} \in \Omega_s\}. \tag{4}$$

Thus, the inference problem consists in the computation of the maximum value of  $\phi$ , and we refer to configurations that adopt this value as *solution configurations*.

**Definition 2.** Let  $(\Phi, D)$  be a valuation algebra induced by a totally ordered idempotent semiring. For  $\phi \in \Phi_s$ , we call  $\mathbf{x} \in \Omega_s$  a *solution configuration* if  $\phi(\mathbf{x}) = \phi^{-s}(\diamond)$ .

*Example 1.* Let  $\phi$  be an semiring valuation defined over propositional variables  $A, B, C$ , and induced by the Tropical Semiring with maximization. We observe the result of full variable elimination:



We see that  $\phi^{-\{A,B,C\}}(\diamond)$  is indeed the maximum value of  $\phi$  and that  $(0_A, 0_B, 1_C)$  and  $(0_A, 1_B, 0_C)$  are its solution configurations.

## 7 Compiling Solution Configurations

This section is dedicated to the task of identifying solution configurations of some joint valuation  $\phi$  that is given by a factorization according to Equation (II). For this purpose, [4] proposed an extension of the fusion algorithm which allows to identify single solution configurations. Whenever a variable is eliminated during the fusion process, this algorithm stores the frame value of the eliminated variable that leads to the maximum value. In this way, solution configurations are obtained at the end of fusion by combining the retained frame values, and from this perspective, the extended fusion algorithm corresponds to *non-serial dynamic programming*. A generalization of this algorithm to valuation algebras induced by idempotent semirings can be found in [13]. This paper embarks on another strategy. Instead of storing partial solution configurations (or frame values) explicitly, we rather represent them in an implicit way by constructing a Boolean function whose set of models corresponds to the solution configurations of  $\phi$ . In short, we *compile* solution configurations into a Boolean function. This idea is sketched in the following example.

*Example 2.* Here, we use propositional formulae as an appropriate representation of Boolean functions. Doing so, the solution configurations found in Example 1 are the models of the Boolean function that corresponds to the following propositional formula:

$$\neg A \wedge ((\neg B \wedge C) \vee (B \wedge \neg C)).$$

We already discussed the semiring that consists of all Boolean functions over a finite set of propositional variables  $r$  with minimization for  $\times$  and maximization for  $+$  respectively. Such a Boolean function  $f$  can likewise be viewed as a set of Boolean vectors  $\mathbf{x} \in \Omega_r$  for which  $f$  evaluates to 1. We refer to these vectors as the *models* of the Boolean function  $f$ . Besides the already introduced constant function  $f_0$  and  $f_1$ , it is very convenient for our purposes to identify the Boolean function that reflects the value of some variable  $X$ . Thus,  $f_X$  represents the Boolean function that evaluates to 1 if and only if variable  $X \in r$  adopts value 1. Accordingly, we write  $f_{\overline{X}}$  to denote its inverse. This notation allows to introduce the concept of *memorizing semiring valuations*.

### 7.1 Memorizing Semiring Valuations

Let  $\mathcal{A} = \langle A, +, \times \rangle$  be a totally ordered idempotent semiring, and  $r$  a finite set of propositional variables. A memorizing semiring valuation  $\phi$  with domain  $s \subseteq r$  is defined to be a function that associates a two-dimensional vector with each configuration  $\mathbf{x} \in \Omega_s$ . The first value of this vector  $\phi_A(\mathbf{x})$  corresponds again to a semiring value, whereas the second  $\phi_F(\mathbf{x})$  constitutes a Boolean function defined over  $r$ . More formally, we have  $\phi : \Omega_s \rightarrow A \times F_r$  and  $\mathbf{x} \mapsto (\phi_A(\mathbf{x}), \phi_F(\mathbf{x}))$  where  $F_r$  denotes the set of Boolean functions over  $r$ . We will again denote the set of all memorizing semiring valuations with domain  $s \subseteq r$  by  $\Phi_s$  and use  $\Phi$  for all memorizing semiring valuations defined over subsets of  $r$ . This definition extends usual semiring valuations by attaching a Boolean function to every configuration  $\mathbf{x} \in \Omega_s$  which will be used during the fusion process to memorize the frame values of eliminated variables that are part of the solution configurations. This is reflected in the following definitions of operations for memorizing semiring valuations:

1. *Labeling*:  $\Phi \rightarrow D$ :  $d(\phi) = s$  if  $\phi \in \Phi_s$ .
2. *Combination*:  $\phi \in \Phi_s, \psi \in \Phi_t$  and  $\mathbf{x} \in \Omega_{s \cup t}$

$$\phi \otimes \psi(\mathbf{x}) = (\phi_A(\mathbf{x}^{1s}) \times \psi_A(\mathbf{x}^{1t}), \min\{\phi_F(\mathbf{x}^{1s}), \psi_F(\mathbf{x}^{1t})\}).$$

Combination of memorizing semiring valuations is defined for both vector components independently. The two values from semiring  $\mathcal{A}$  are again combined by the corresponding generic semiring operation  $\times$ , and the same holds for the Boolean functions because in this semiring  $\times$  corresponds to minimization. The semantics of this definition is that in case of combination, we combine conjunctively the memories of both factors  $\phi$  and  $\psi$ .

3. *Variable Elimination*:  $\phi \in \Phi_s, Y \in s$  and  $\mathbf{x} \in \Omega_{s - \{Y\}}$

$$\phi^{-Y}(\mathbf{x}) = (\phi_A^{-Y}(\mathbf{x}), \phi_F^{-Y}(\mathbf{x}))$$

where

$$\phi_A^{-Y}(\mathbf{x}) = \phi_A(\mathbf{x}, 0_Y) + \phi_A(\mathbf{x}, 1_Y)$$

and

$$\phi_F^{-Y}(\mathbf{x}) = \begin{cases} \min\{f_Y, \phi_F(\mathbf{x}, 1_Y)\}, & \text{if } \phi_A(\mathbf{x}, 1_Y) >_{id} \phi_A(\mathbf{x}, 0_Y), \\ \min\{f_{\overline{Y}}, \phi_F(\mathbf{x}, 0_Y)\}, & \text{if } \phi_A(\mathbf{x}, 1_Y) <_{id} \phi_A(\mathbf{x}, 0_Y), \\ \max\{\min\{f_Y, \phi_F(\mathbf{x}, 1_Y)\}, \min\{f_{\overline{Y}}, \phi_F(\mathbf{x}, 0_Y)\}\}, & \text{otherwise.} \end{cases}$$

We again define this operation per component. For the two values of semiring  $\mathcal{A}$  we proceed as normal and apply the corresponding semiring addition. In contrast, the computation of the Boolean function varies with respect to the order of the two semiring values  $\phi_A(\mathbf{x}, 0_Y)$  and  $\phi_A(\mathbf{x}, 1_Y)$ , and this is where we memorize the frame value of  $Y$  that is contained in a solution configuration. For example, if  $\phi_A(\mathbf{x}, 1_Y) >_{id} \phi_A(\mathbf{x}, 0_Y)$ , then  $Y = 1$  must hold in every solution configuration of  $\phi$ . This is guaranteed by the particular Boolean function  $f_Y$  that is conjunctively added to the memory  $\phi_F(\mathbf{x}, 1_Y)$ , which contains the constraints from former eliminations. A similar reasoning applies in the case where  $\phi_A(\mathbf{x}, 0_Y) >_{id} \phi_A(\mathbf{x}, 1_Y)$ . Finally, if the two values  $\phi_A(\mathbf{x}, 0_Y)$  and  $\phi_A(\mathbf{x}, 1_Y)$  are equal, both constructions are combined disjunctively, because the solution configurations do not depend on the value  $Y$ .

**Theorem 2.** *A system of memorizing semiring valuations with labeling, combination and marginalization as defined above, satisfies the axioms of a valuation algebra.*

This theorem is proved in [13] and its statement allows to apply local computation, or more concretely the fusion algorithm on memorizing semiring valuations. Thus, for a given factorization  $\phi = \phi_1 \otimes \dots \otimes \phi_n$  of semiring valuations over a totally ordered, strictly monotonic, idempotent semiring  $\mathcal{A}$ , we embed the factors  $\phi_i$  into a set of memorizing semiring valuations as follows: For  $i = 1, \dots, n$  we define

$$\widehat{\phi}_i : \mathbf{x} \in \Omega_{d(\phi_i)} \mapsto (\phi_i(\mathbf{x}), f_1).$$

After this initialization step, we execute the fusion algorithm and eliminate all variables. We obtain

$$\widehat{\phi}^{-s}(\diamond) = \left( \widehat{\phi}_1 \otimes \dots \otimes \widehat{\phi}_n \right)^{-s}(\diamond).$$

Clearly, the semiring component  $\widehat{\phi}_A^{1\emptyset}(\diamond)$  contains again the maximum value of  $\phi$  over all its configurations, i.e.

$$\widehat{\phi}_A^{-s}(\diamond) = \phi^{-s}(\diamond) = \max\{\phi(\mathbf{x}), \mathbf{x} \in \Omega_s\}.$$

Let us now focus on the Boolean function  $\widehat{\phi}_F^{-s}(\diamond)$  that has been built simultaneously. The following theorem confirms what we have foreshadowed all along, namely that the set of models of this function corresponds exactly to the solution configurations we are looking for.

**Theorem 3.** *For  $\mathbf{x} \in \Omega_s$  and  $s = d(\phi)$  we have*

$$\left( \widehat{\phi}_F^{-s}(\diamond) \right) (\mathbf{x}) = 1 \text{ if and only if } \phi(\mathbf{x}) = \phi^{-s}(\diamond).$$

Thus, every solution configuration of  $\phi$  evaluates the constructed Boolean function to 1 and is therefore a model. Conversely, every model is also a solution configuration of  $\phi$ . The proof of this result is given in [13].

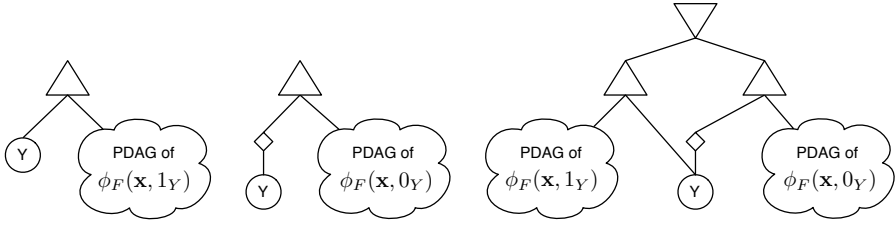


Fig. 1. Variable elimination rules of memorizing semiring valuations as PDAGs

### 7.2 Representing Boolean Functions

Naturally, an indispensable requirement for the applicability of this theory is that the models of the final Boolean function can be enumerated efficiently, i.e. in polynomial time. This depends first and foremost on a suitable representation of this particular Boolean function. Here, we propose its representation as *propositional directed acyclic graph* (PDAG) [9].

**Definition 3.** A PDAG over the set of propositional variables  $r$  is a rooted directed acyclic graph of the following form:

1. Leaves are represented by  $\circ$  and labeled with  $\top$  (true),  $\perp$  (false), or  $X \in r$ .
2. Non-leaves are represented by  $\triangle$  (logical conjunction),  $\nabla$  (logical disjunction) or  $\diamond$  (logical negation).
3.  $\triangle$ - and  $\nabla$ -nodes have at least one child and  $\diamond$ -nodes have exactly one child.

Figure 1 illustrates the PDAG structures that are created from the three variable elimination rules of memorizing semiring valuations. Combination on the other hand just connects existing PDAGs by a conjunction node to a new PDAG. Thus, because all variables are eliminated, we obtain at the end of the fusion algorithm a single PDAG structure that represents  $\widehat{\phi}_F^{-s}(\diamond)$ . Some particularities of this graph are summarized in Lemma 2.

**Lemma 2.** The PDAG representation of  $\widehat{\phi}_F^{-s}(\diamond)$  satisfies the following properties:

1. *Simple-Negation:* Each child of a  $\diamond$ -node is a leaf.
2. *Decomposability:* The sets of variables that occur in the sub-trees of every  $\triangle$ -node are disjoint.
3. *Determinism:* The children of every  $\nabla$ -node are pairwise logically contradictory, i.e. if  $\alpha_i$  and  $\alpha_j$  are two children of the same  $\nabla$ -node, we have  $\alpha_i \wedge \alpha_j \equiv \perp$ .

The first property follows directly from PDAGs 2 and 3 in Figure 1 because these are the only rules that create negation nodes. A variable node is created whenever the corresponding variable is eliminated. Hence, every variable node occurs exactly once in this PDAG which proves Property 2. Finally, we can conclude from PDAG 3 in Figure 1 that in every model of the disjunction node’s left child,  $Y = 1$  must hold. Similarly,  $Y = 0$  must hold in the right child and therefore, the two model sets are disjoint. This is the

statement of Property 3. A PDAG that satisfies all three properties of Lemma 2 is called *cdn-PDAG* [9] or *d-DNNF* [8] and it turns out that model enumeration can indeed be done in polynomial time upon such PDAGs. A corresponding algorithm for this task is given in [17].

It is worth mentioning that during the fusion process, d-DNNFs are built from connecting existing d-DNNFs by either a conjunction or a disjunction node. However, we know from [8] that the d-DNNF language is not closed under these operations. This means concretely that it is in general not possible to reconstruct a d-DNNF structure from the conjunction or disjunction of two d-DNNFs in polynomial time. Fortunately, this does not hold for the case at hand. Since these constructions are performed as valuation algebra operations, we directly obtain the *cdn*-properties whenever we join two existing d-DNNFs by the rules specified for memorizing semiring valuations. This features our approach in contrast to similar techniques where the needed graph properties have to be reestablished [7].

### 7.3 Further Efficient Queries

Besides efficient model enumeration, d-DNNFs allow many other *queries* to be performed efficiently, and this fact constitutes the worth of our method compared with classical approaches that essentially content themselves with model identification. [9] give a comprehensive listing of such queries whose complete disquisition would go beyond the scope of this article. Nevertheless, we will suggest some of these possibilities in order to point out the potential of our approach in view of relevant applications in diagnosis for example.

- *Counter-Model Enumeration*: The d-DNNF constructed by the fusion algorithm allows also to enumerate all configurations of  $\phi$  that are not solution configurations, i.e. all configurations  $\mathbf{x} \in \Omega_s$  with  $\phi(\mathbf{x}) <_{id} \phi^{-s}(\diamond)$ .
- *Model Counting*: Counting the number of solution configurations can also be done efficiently.
- *Validity*: This answers the query if  $\widehat{\phi}_F^{-s}(\diamond) \equiv f_1$ , i.e. if all configurations of  $\phi$  adopt the same value.
- *Probability Computation*: If we assume independent marginal probabilities  $p(X)$  for all variables  $X \in s$ , we can efficiently evaluate the probability of the Boolean function  $p(\widehat{\phi}_F^{-s}(\diamond))$ .
- *Probabilistic Equivalence Test*: If two different factorizations over the same set of variables are given, d-DNNFs allow to test probabilistically if the two joint valuations  $\phi_1$  and  $\phi_2$  adopt the same solution configurations, i.e. for all  $\mathbf{x} \in \Omega_s$ ,  $\phi_1(\mathbf{x}) = \phi^{-s}(\diamond)$  if and only if  $\phi_2(\mathbf{x}) = \phi^{-s}(\diamond)$ .

Although this is only a small selection from the extensive list of queries that can be performed efficiently on d-DNNFs and therefore on the fusion algorithm's result, we come to the conclusion that the representation of solution configurations by a Boolean functions offers a lot more possibilities for further evaluations than traditional approaches. Furthermore, because this method is also based on local computation, no loss of efficiency occurs.

## 8 Conclusion

With the (extended) fusion algorithm, we are in possession of an efficient tool to compute solution configurations of optimization problems that are given as factorizations of semiring valuations. However, the requirements of diagnosis are rarely limited to the identification of solution configurations. Moreover, we often want to perform some further evaluations of these solution configurations without their explicit enumeration. This paper proposes therefore to compile the solution configuration set into a Boolean function and by use of current knowledge compilation techniques, we obtain a very compact, graphical representation of the solution configuration set that allows to carry out efficiently a large collection of new queries, including the enumeration of solution configurations. Additionally, this compilation process does not forfeit efficiency because it is based on the same local computation scheme. This features our new approach which is furthermore a delightful combination of results from different fields of AI research.

## References

1. Shenoy, P.P., Shafer, G.: Axioms for probability and belief-function propagation. In: Shachter, R.D., Levitt, T.S., Kanal, L.N., Lemmer, J.F. (eds.) *Uncertainty in Artificial Intelligence 4. Machine intelligence and pattern recognition 9*, 169–198 (1990)
2. Shafer, G.: *An axiomatic study of computation in hypertrees*. Working Paper 232, School of Business, University of Kansas (1991)
3. Kohlas, J.: *Information Algebras: Generic Structures for Inference*. Springer, Heidelberg (2003)
4. Shenoy, P.: Axioms for dynamic programming. In: Gammerman, A. (ed.) *Computational Learning and Probabilistic Reasoning*, pp. 259–275. Wiley, Chichester (1996)
5. Bertele, U., Brioschi, F.: *Nonserial Dynamic Programming*. Academic Press, London (1972)
6. Wilson, N.: Decision diagrams for the computation of semiring valuations. In: *IJCAI 2005. 19th International Joint Conference on Artificial Intelligence*, pp. 331–336 (2005)
7. Mateescu, R., Dechter, R.: Compiling constraint networks into and/or multi-valued decision diagrams (aomdds). In: Benhamou, F. (ed.) *CP 2006. LNCS, vol. 4204*, pp. 329–343. Springer, Heidelberg (2006)
8. Darwiche, A., Marquis, P.: A knowledge compilation map. *J. Artif. Intell. Res (JAIR)* 17, 229–264 (2002)
9. Wachter, M., Haenni, R.: Propositional DAGs: a new graph-based language for representing Boolean functions. In: *KR 2006. 10th International Conference on Principles of Knowledge Representation and Reasoning*, pp. 277–285 (2006)
10. Shenoy, P.: Valuation-based systems: A framework for managing uncertainty in expert systems. In: *Fuzzy Logic for the Management of Uncertainty*, pp. 83–104 (1992)
11. Schneuwly, C., Pouly, M., Kohlas, J.: *Local computation in covering join trees*. Technical Report 04-16, University of Fribourg (2004)
12. Schweizer, B., Sklar, A.: Statistical metric spaces. *Pacific J. Math.* 10, 313–334 (1960)
13. Pouly, M., Kohlas, J.: *Local computation & dynamic programming*. Technical Report 07-02, University of Fribourg (2007)

14. Kohlas, J., Wilson, N.: Exact and approximate local computation in semiring induced valuation algebras. Technical Report 06-06, University of Fribourg (2006)
15. Bistarelli, S., Montanari, U., Rossi, F., Schiex, T., Verfaillie, G., Fargier, H.: Semiring-based CSPs and valued CSPs: Frameworks, properties, comparison. *Constraints* 4, 199–240 (1999)
16. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* 1 (1978)
17. Darwiche, A.: Decomposable negation normal form. *J. ACM* 48, 608–647 (2001)

# Implementing Knowledge Update Sequences

Juan C. Acosta Guadarrama\*

Institute of Computer Science,  
TU-Clausthal, Germany  
guadarrama@in.tu-clausthal.de

**Abstract.** Update of knowledge bases is becoming an important topic in Artificial Intelligence and a key problem in knowledge representation and reasoning. One of the latest ideas to update logic programs is choosing between models of Minimal Generalised Answer Sets to overcome disadvantages of previous approaches. This paper describes an implementation of the declarative version of updates sequences that has been proposed as an alternative to syntax-based semantics. One of the main contributions of this implementation is to use DLV's Weak Constraints to compute the model(s) of an update sequence, besides presenting the precise definitions proposed by the authors and an online solver. As a result, the paper makes an outline of the basic structure of the system, describes the employed technology, discusses the major process of computing the models, and illustrates the system through examples.

**Keywords:** Answer Set Programming, Knowledge Representation, program transformation, implementation, non-monotonic reasoning, belief revision.

## 1 Introduction

As one of the major and traditional topics of Artificial Intelligence over the last years, knowledge representation and reasoning has proved to be a strong theoretical framework for Logic Programming to manage knowledge bases. As a result, this particular topic has become more widely applied in the administration knowledge bases of intelligent (rational) agents, especially when talking about an agent's incomplete knowledge in a changing environment. In particular, this area of research is known in the literature as *belief updates*.

A traditional and general goal of belief updates is dealing with contradictory information or with new data. However, there are particular rare (and possible) challenging situations that might lead to *counterintuitive* models of the environment that have been subject of recent research and matter of formulation of new principles.

Several authors like [EFST02](#), [SI03](#), [ZF05](#), [ABBL05](#) make the foundation of their approaches on *Answer Set Programming* [GL88](#) —or simply ASP— for being one of the most solid and studied semantics for logic programs over the

---

\* This project is mainly supported by a CONACYT Doctorate Grant.



last years. However, most of them are founded on a *causal rejection principle* [ABBL05, EFST02] that leads to unintuitive behaviour under certain circumstances [ABBL05, AGDOZ05]. In order to illustrate this claim, consider the following theory, proposed to solve a very similar problem in [ABBL05], describing some beliefs about the sky.

*Example 1*

$$\begin{array}{ll} \Pi_1 = \{ \text{day} \leftarrow \text{not night} & \text{night} \leftarrow \text{not day} \\ \text{stars} \leftarrow \text{night, not cloudy} & \neg \text{stars} \leftarrow \top \} \end{array}$$

whose unique answer set is  $\{\text{day}, \neg \text{stars}\}$ . If this is updated with the following program

$$\Pi_2 = \{ \text{stars} \leftarrow \text{constellations} \quad \text{constellations} \leftarrow \text{stars} \}$$

one can realise that  $\Pi_2$  contains only one new extra atom (*constellations*) with respect to the original program  $\Pi_1$ , that in this case it is considered a synonymous of *stars*. So, it should not affect the original knowledge base. However, as first pointed out by [EFST02, ABBL05],  $\Pi_2$  introduces a new answer set in at least those semantics based on the causal rejection principle that does not seem intuitive:  $\{\text{stars}, \text{constellations}, \text{night}\}$ .

One of the solutions comes from [AGDO06] who propose a semantics based upon *Minimal Generalised Answer Sets* [KM90, BG03] —or MGAS hereafter— that satisfies several structural properties, overcoming the above problems from other proposals. In contrast to other some few approaches with implementation<sup>1</sup> such a proposal lacks of a solver, needed as one of the main components of logic programming to verify, confirm, test, program and compare with accuracy, as well as to be a key component for more complex AI applications.

In this paper, we present a general description of a system to implement update sequences as proposed in [AGDO06], as well as tools, methods, and directions to get the running solver itself, as well as some minor amends to their original definitions that provide more precision. In general, the paper is organised as follows. It starts with preliminary notation in Section 2, followed by the main mended definitions of the semantics (Section 3) and the core of the paper in Section 4.

## 2 Preliminaries

This section is quite general and then it expects the reader to be familiar with basic notions of logic programming and non-monotonic reasoning from the literature.

<sup>1</sup> See for example the implemented versions of [ABBL05, EFST02] at [centria.di.fct.unl.pt/~banti/FedericoBantiHomepage/refdplp.htm](http://centria.di.fct.unl.pt/~banti/FedericoBantiHomepage/refdplp.htm) and [www.kr.tuwien.ac.at/staff/giuliana/project.html#Download](http://www.kr.tuwien.ac.at/staff/giuliana/project.html#Download), respectively.

### 2.1 Logic Programming and Answer Sets

The main base of our proposal is Answer Set Programming —ASP— [GL88] that is a well-known semantics for its logical characteristics, for its intuitive language to represent non-monotonic knowledge bases, and it is one of the foundations the authors in [AGDO06] propose for their approach. Its formal language and some more notation are introduced *very* briefly as follows.

**Definition 1 (Language  $\mathcal{L}_{ASP}$  of logic programs).** *In the following we use the language of propositional logic with propositional symbols:  $p_0, p_1, \dots$ ; connectives: “ $\wedge$ ” (conjunction), “ $\vee$ ” (disjunction), “ $\leftarrow$ ” (implication), “ $\perp$ ” (falsum), “ $\top$ ” (verum), “not” (default negation), “ $\neg$ ” (strong negation); auxiliary symbols: “(”, “)” (parentheses). The propositional symbols are also called atoms or atomic propositions. A literal is an atom or an atom negated by  $\neg$ . A rule  $\rho$  is an ordered pair  $H(\rho) \leftarrow B(\rho)$ , where  $H(\rho)$  is a literal or null, and  $B(\rho)$  a finite set of literals or default-negated literals.*

For a literal  $l$ , the *complementary literal*,  $\neg l$ , is  $\neg p$  if  $l = p$ , and  $p$  if  $l = \neg p$ , for some atom  $p$ . Given a set of atoms  $A$ , we denote by  $\text{not } A = \{\text{not } a \mid a \in A\}$ ;  $\neg A = \{\neg a \mid a \in A\}$ ; given another set  $M \subseteq A$ , we define  $\widetilde{M} = A \setminus M$ ; finally, a signature  $\Sigma_{\Pi}$  as the finite set of atoms occurring in  $\Pi$  and  $\text{Lit}_S$  the set  $S \cup \neg S$  for all literals over  $S$ .

The semantics of such programs, as defined in the literature, consists of reducing the general rules to rules without default negation “not”, because the latter are universally well understood. For space constraints, we just mention it as follows.

**Definition 2 (Answer Sets [GL88]).** *A set of literals  $X$  is an answer set of  $\Pi$  if, by definition,  $X = \text{Min}(\Pi^X)$ .*

which can be computed with DLV.

**Definition 3 (Extended Disjunctive Logic Program EDLP [GL91]).** *An extended disjunctive logic program is a set of rules of the form*

$$p_0 \mid p_1 \mid \dots \mid p_l \leftarrow q_1, \dots, q_m, \text{not } q_{m+1}, \dots, \text{not } q_n \tag{1}$$

where  $p_i$  and  $q_i$  are literals and  $l, m, n > 0$ .

As a main component of our solver, a weak constraint is a constraint that may be violated in order to establish priorities among models, which was introduced in [LPF+06] having the following form.

**Definition 4 (Weak Constraint [LPF+06]).** *Weak constraint ( $wc$ ) is an expression of the form*

$$:\sim b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m [w : l] \tag{2}$$

where for  $0 \leq k \leq m$ ,  $b_1, \dots, b_m$  are literals, while  $w$  (the weight) and  $l$  (the level, or layer) are positive integer constants or variables. For convenience,  $w$

and/or  $l$  may be omitted and are set to 1 in this case. The sets  $B(w)$ ,  $B^+(w)$ , and  $B^-(w)$  of a weak constraint  $w$  are defined in the same way as for regular integrity constraints.

and, out of the answer sets of a program, the interpretations consist of minimising the sum of weights of violated weak constraints in the highest priority level, and among them those which minimise the sum of weights of the violated weak constraints in the next lower level, and so on.

**Definition 5** ([LPF<sup>+</sup>06]). *Given a ground program  $\Pi$  with weak constraints  $WC(\Pi)$ , the objective function  $H^\Pi(\mathcal{S})$  for  $\Pi$  and an answer set  $\mathcal{S}$  is defined by using an auxiliary function  $f_\Pi$  that maps levelled weights to weights without levels:*

$$\begin{aligned} f_\Pi(1) &= 1 \\ f_\Pi(n) &= f_\Pi(n-1) \cdot |WC(\Pi)| \cdot w_{max}^\Pi + 1, n > 1 \\ H^\Pi(\mathcal{S}) &= \sum_{i=1}^{l_{max}^\Pi} (f_\Pi(i) \cdot \sum_{w \in N_i^\Pi(\mathcal{S})} weight(w)) \end{aligned}$$

where  $w_{max}^\Pi$  and  $l_{max}^\Pi$  denote the maximum weight and maximum level over the weak constraints in  $\Pi$ , respectively;  $N_i^\Pi(\mathcal{S})$  denotes the set of the weak constraints in level  $i$  that are violated by  $\mathcal{S}$ , and  $weight(w)$  denotes the weight of the weak constraint  $w$ .

The reader should note that  $|WC(\Pi)| \cdot w_{max}^\Pi + 1$  is greater than the sum of all weights in the program, and therefore guaranteed to be greater than the sum of weights of any single level [LPF<sup>+</sup>06].

Although ASP is our main base, we need an intermediate means to set up preferences amongst models, so that we may choose the ones according to general principles and postulates. One of such intermediate mechanisms was introduced as Abductive Logic Programming in [KM90] and is briefly presented in the following section.

## 2.2 Minimal Generalised Answer Sets

As one of the semantics to interpret abductive programming, Minimal Generalised Answer Sets (MGAS) provides a more general and flexible semantics than standard ASP. Some of the founding definitions are as follows.

**Definition 6** (Abductive Logic Program [KM90]). *An abductive logic program is a pair  $\langle \Pi, \mathcal{A} \rangle$  where  $\Pi$  is an arbitrary program and  $\mathcal{A}$  a set of literals, called abducibles.*

Afterwards, a way to interpret an abductive program is by its generalised answer sets. The general intuition behind this process consists of merging combinations of abducibles with the original program. Then, the resulting programs are interpreted under answer sets semantics, as formally expressed in the following definition.

**Definition 7 (Generalised Answer Sets GAS [KM90]).**  $M(\Delta)$  is a generalised answer set of the abductive program  $\langle \Pi, \mathcal{A} \rangle$  iff  $\Delta \subseteq \mathcal{A}$  and  $M(\Delta)$  is an answer set of  $\Pi \cup \{H \leftarrow \top \mid H \in \Delta\}$ .

Once we get more than one generalised answer set, a preferred order can be chosen over their inclusion order.

**Definition 8 (Abductive Inclusion Order [KM90]).** An order over generalised answer sets is as follows: Let  $M(\Delta_1)$  and  $M(\Delta_2)$  be generalised answer sets of  $\langle \Pi, \mathcal{A} \rangle$ , we define  $M(\Delta_1) \leq_{\mathcal{A}} M(\Delta_2)$  iff  $\Delta_1 \subseteq \Delta_2$ .

Hereafter, we introduce a more readable notation to represent GAS's by  $M_{\Delta}$ , where  $M$  is the resulting answer set with the abducible set  $\Delta$ .

### 3 Update Operation

Intuitively a sequence of programs is the relaxation of all rules in previous programs to the current evolving state, with a unique abducible. As a result, there is a transformed *relaxed* program as a part of an abducible program, and the other part is the set of abducibles. We obtain the corresponding GAS's of the abductive program and choose the minimal with respect to its sequence order and to its inclusion. Finally, we intersect the MGAS with the original alphabet in order to filter out the abducibles.

**Definition 9 (Relaxed Rule, Sequence, Program [AGDO06]).** Let  $\Pi = \Pi_1, \dots, \Pi_n$  be a sequence of extended logic programs in the language  $\mathcal{L}_{ASP}$  over the set of atoms  $A$  and  $n \geq 2$ .

- Given a rule  $\rho \in \mathcal{L}_{ASP}$ , and a new distinguished atom  $\alpha \notin A$ , define the  $\rho$ -relaxed form by  $\text{Head}(\rho) \leftarrow \text{Body}(\rho) \cup \{\text{not } \alpha\}$ .
- We define the relaxed sequence of  $\Pi$  as  $\Pi'_1, \dots, \Pi'_{n-1}$  where each rule  $\rho' \in \Pi'_i$  is the  $\rho$ -relaxed form of a rule  $\rho \in \Pi_i$  by a unique atom  $\alpha \notin A$ , and  $1 \leq i \leq n - 1$ .
- Given the relaxed sequence  $\Pi' = \Pi'_1, \dots, \Pi'_{n-1}$  of  $\Pi$ , we define its relaxed program  $\Pi'$  as the set of all the rules in the relaxed sequence:  $\Pi' = \Pi'_1 \cup \dots \cup \Pi'_{n-1}$

**Definition 10 (Relaxed-Sequence Order, Update Order [AGDO06]).** Let  $\Pi = \Pi_1, \dots, \Pi_n$  be an update sequence with  $n \geq 2$ ; and let  $\Pi' = \Pi'_1, \dots, \Pi'_{n-1}$  be its corresponding relaxed sequence, with  $\Pi'$  as its corresponding relaxed program; let  $\Pi'_i, \Pi'_j$  be two arbitrary relaxed programs in the relaxed sequence with  $1 \leq i \leq j \leq n - 1$ ; let  $M(\Delta_1)$  and  $M(\Delta_2)$  be generalised answer sets of  $\langle \Pi' \cup \Pi_n, \mathcal{A} \rangle$ .

- $M(\Delta_1) \leq_S M(\Delta_2)$  iff there is an abducible  $\alpha_2 \in \Delta_2$  such that, for every  $\alpha_1 \in \Delta_1$  the following condition holds:  $\alpha_1 \in \mathcal{L}_{\Pi'_i}$  and  $\alpha_2 \in \mathcal{L}_{\Pi'_j}$ .
- $M(\Delta_1) \leq_U M(\Delta_2)$  iff  $M(\Delta_1)$  and  $M(\Delta_2)$  are MGAS's and  $M(\Delta_1) \leq_S M(\Delta_2)$ .

Intuitively, there is an order with respect to the latest update —which corresponds to postulates (R1) and (U1) in [KM91b, KM91a]—, and with respect to a minimal change: MGAS.

With this order, we can establish a variant of MGAS with respect to update order as follows.

**Definition 11 (Update MGAS, UMGAS [AGDO06]).**  *$M(\Delta)$  is an update minimal generalised answer set (written UMGAS) of  $\langle \Pi, \mathcal{A} \rangle$  iff  $M(\Delta)$  is a generalised answer set of  $\langle \Pi, \mathcal{A} \rangle$  and it is minimal w.r.t. its abductive update order.*

Now, the process of updating a sequence of logic programs consists of transforming the update sequence into a single abductive program. Formally,

**Definition 12 (Update Program [AGDO06]).** *Given a sequence of update extended logic programs  $\Pi_{\otimes} = \Pi_1 \otimes \dots \otimes \Pi_n$ , with  $n \geq 2$ , over a set of atoms  $A$  and its corresponding relaxed program  $\Pi'$ , its update program is the abductive program  $\langle \Pi' \cup \Pi_n, \mathcal{A} \rangle$  where  $\mathcal{A}$  is the set of abducibles, such that  $A \cap \mathcal{A} = \emptyset$ , and  $\otimes$  is our update operator.*

We choose among its generalised answer sets, with respect to their update order, and filter out the abducibles.

**Definition 13 (Update Answer Set [AGDO06]).** *Let  $\Pi_{\otimes}$  be an update sequence over a set of atoms  $A$ . Then,  $S \subseteq A$  is an update answer set of  $\Pi_{\otimes}$  iff  $S = S' \cap A$  for some UMGAS  $S'$  of  $\Pi_{\otimes}$ .*

Further theoretical results both on the properties of DLV's weak constraints and on the slight amends to the original definition are omitted due to page-limit restrictions.

## 4 Implementation

Currently there are two major efficient solvers for ASP with a long background of implementation and research. Namely, DLV [LPF+06] and SMOBELS [NS97], and our system is at a higher level because it updates ELP programs.

One of the proposals [AGDOZ05] to implement updates in MGAS was a setting of preferred disjunctive logic programs in ODLP [Bre02] and has an implementation for pairs of programs at [www2.in.tu-clausthal.de/~guadarrama/updates/fuhrmann/process.php](http://www2.in.tu-clausthal.de/~guadarrama/updates/fuhrmann/process.php). However, the system as well as the semantics it implements, is limited to pairs of programs. The justification in [AGDOZ05] to use ODLP is the implemented solver called PSMODELS<sup>2</sup> that is an extension to SMOBELS [NS97] to compute preferred answer sets. Unfortunately, up to now there is no stable version and the current one (v. 2.26a) endures some few bugs<sup>3</sup>. Moreover, it is believed that DLV significantly outperforms SMOBELS [LPF+06],

<sup>2</sup> <http://www.tcs.hut.fi/Software/smodels/priority/>

<sup>3</sup> Try to compute the preferred models of the simple program like  $\{a\}$ .

not to mention that ODLP is such a colossal system that can do much more complex tasks than just minimal inclusion models.

As a result we propose the use of Weak Constraints in DLV for their characteristics of minimality under set inclusion [LPF<sup>+</sup>06], and that enjoy the above benefits of being in DLV with no more extra throughput added to the task.

#### 4.1 The Parser

Differently from the implementation of [AGDOZ05], which has parser embedded in its PHP [4] code, this new parser has been compiled in C on the MacOS X<sup>TM</sup> platform at Darwin<sup>TM</sup> level, and it should be available for Linux at the same location, as an alternative. The advantage of having a UNIX binary module is the ease to be plugged in to other modules so as to form a more complex application.

MacOS X<sup>TM</sup>/Darwin<sup>TM</sup> is a BSD branch of UNIX, that has a ported set of *Lex* and *Yacc* utilities in their GNU versions of *Flex* and *Bison*, respectively. *Flex* is a short name for Fast Lexical Analyser that generates code to scan text through regular expressions pattern matching.

The following tokens are implemented for this update solver.

NAME	[[:alnum:]-]+
PnSTART	"{"
PnEND	"}"
SNOT	["~]
GETS	":-"
NOT	"not"
RULEEND	". "
CONJUNCT	","
DISJUNCT	" "

In order to avoid confusion in human reading, dash "-" cannot be part of a name. Other tokens like PnSTART and PnEND split the sequence of programs into individual programs, while the rest of the tokens need no explanation.

As soon as *Flex* decomposes the text of a program sequence, its output is taken on by *Yacc*, which gives a meaning to each correct structure of rules and program sequence. In particular, the *Yacc* process specifies the grammar for update sequences introduced in [AGDO06][5]. It also weakens each rule in all programs of the sequence with a new unique atom and establishes a preference relation among such atoms, according to the sequence the rule is in. Last, this process is responsible of an error-checking mechanism that verifies the correctness of the program according to the BNF grammar bellow.

```
<sequence> ::= <sequence> '{' <program> '}'
<program> ::= <program> <rule>
```

<sup>4</sup> This is a script language quite suitable for small processes of dynamic contents on web pages.

<sup>5</sup> The minor changes to the original formulation leave the original main idea unchanged and correct typos that may be easily noticed in Definition [10].

```

<rule>      ::= <head> END
              | <head> GETS <body> END
<head>      ::= <POST>
<POST>      ::= <LITERAL>
              | <POST> DISJUNCT <LITERAL>
<body>      ::= <PRE>
<PRE>       ::= <LITERAL>
              | NOT <LITERAL>
              | <PRE> CONJUNCT <LITERAL>
              | <PRE> CONJUNCT NOT <LITERAL>
<LITERAL>   ::= <ATOM>
              | SNOT <ATOM>
<ATOM>      ::= NAME
              | NAME ' ( ' NAME ' , ' NAME ' ) '

```

As before mentioned, once a rule is analysed and weakened, the process constructs a pair of new rules with the weakening atoms under weak constraints semantics [CDE<sup>+</sup>02, LPP<sup>+</sup>06] in the following form

$$- \alpha_i \mid \alpha_i \leftarrow \top \quad (3)$$

$$:\sim \alpha_i [p : p] \quad (4)$$

where  $i$  represents the  $i$ -th abducible  $\alpha$  and  $p$  the  $p$ -th program, the latter forming a  $[weight : level]$  weak constraint, where  $weight = level$ .

The intuition behind this formulation is computing the MGAS of the abductive program by violating the least number of weak constraints. In addition, the  $[p : p]$  relation represents the sequence order from Definition 10. That is to say, the models are chosen amongst those that violate the least number of weak constraints with the least weight-level.

*Example 2.* Suppose the sequence

$$\begin{aligned} \Pi_1 &: \{a \leftarrow \text{not } b\} \\ \Pi_2 &: \{b \leftarrow \text{not } a \quad c \leftarrow a \quad d \leftarrow \neg a, b\} \\ \Pi_3 &: \{e \leftarrow \text{not } b\} \end{aligned}$$

The corresponding abductive program is  $\langle \Pi' \cup \Pi_3, \mathcal{A}^* \rangle$  where

$$\begin{aligned} \Pi' = \{ &a \leftarrow \text{not } b, \text{not } \alpha_1 && b \leftarrow \text{not } a, \text{not } \alpha_2 \\ &c \leftarrow a, \text{not } \alpha_3 && d \leftarrow \neg a, b, \text{not } \alpha_4 && e \leftarrow \text{not } b\} \end{aligned}$$

and  $\mathcal{A}^* = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ . Such an abductive program is transformed into a preference program as

```

a:- not b, not alpha_1.
-alpha_1 | alpha_1.
~ alpha_1. [1:1]

```

```

b:- not a, not alpha_2.
-alpha_2 | alpha_2.
~ alpha_2. [2:2]
c:- a, not alpha_3.
-alpha_3 | alpha_3.
~ alpha_3. [2:2]
d:- -a, b, not alpha_4.
-alpha_4 | alpha_4.
~ alpha_4. [2:2]
e:- not b, not alpha_5.
-alpha_5 | alpha_5.
~ alpha_5. [3:3]

```

## 4.2 The Top Module

The top module consists of the display of the original sequence, its transformation to abductive program, as well as the result of interpreting such an abductive program under MGAS and the update answer sets of the sequence. All in all are coded in a UNIX script, with some simple sub-processes that filter in the needed text from the formatted output of DLV. Last, this main module is also responsible of dealing with the user interface in HTML, by getting the input sequence in a text pane on a web page and processing it to display the output within a new web page.

**The Abductive Program.** The abductive program is indeed coded into a preference relation of weak constraints, consisting of the weakened program where each rule has its corresponding pair of disjunctive abducibles (3) and a weak constraint (4).

This simple process forms the triple rule at the parsing stage by keeping a counter for each abducible and for each program, which is printed out once a rule is recognised: the relaxed rule (Definition 9), the disjunctive rule (3) and the weighted weak constraint (4).

**Computing UMGAS's.** Computing UMGAS's is a straightforward process that takes the abductive preference program from the previous process as an input and passes it on to DLV [LPF<sup>+</sup>06] solver. The general intuition behind this (optimised) solver is computing the reduct (Definition 2) of the input ELP program that returns zero or more answer sets. Then, it chooses the best model(s), according to the weak constraints —Definition 4. As mentioned before, the best answer set(s) are those that violate the least number of the least weighted/levered weak constraints —Definition 4, 5.

From Example 2, the system returns the following two UMGAS's:

$$\{b\}_{\{-\alpha_1, -\alpha_2, -\alpha_3, -\alpha_4, -\alpha_5\}}, \{a, c, e\}_{\{-\alpha_1, -\alpha_2, -\alpha_3, -\alpha_4, -\alpha_5\}}$$

**The Update Answer Sets.** Finally, the last process is a simple filtering with UNIX processes of the abductive atoms that removes them from the output and gives the desired result. In this example, just  $\{b\}, \{a, c, e\}$ .



### 4.3 Experimental Results

The system at [www2.in.tu-clausthal.de/~guadarrama/updates/seqs.html](http://www2.in.tu-clausthal.de/~guadarrama/updates/seqs.html) runs online with some few mirror sites shown there. The binaries for command-line process may be downloaded up to now for MacOS X<sup>TM</sup> Darwin<sup>TM</sup> on PowerPC<sup>TM</sup> as well as for Linux on Intel® from the same sites.

## 5 Discussion and Future Work

Besides some minor amends to the original definitions of update sequences to make them precise, we have presented general methods for rapid prototyping of logic programming semantics and for further research in optimisation techniques, and implemented the declarative version of both an update semantics and MGAS's. The system has been developed with strong emphasis in declarative programming, in just some few fragments of procedural modules, in order to make it easily modifiable for particular frameworks and as an evidence to confirm claims of the original semantics here implemented. Another of its highlights is its modularity and UNIX philosophy that allows it to be a web service and easily plugged in to other systems via on-line even without needing to download it. Moreover, its simple standard graphical user interface in HTML makes it very easy to use, compared to most of the solvers implemented for command-line use.

As one of the main components of Logic Programming, implementation of semantics helps quickly understand it (for educational proposes and for a reliable comparison tool, for instance), spread it and compute large knowledge bases for more complex applications and future frameworks.

## References

- [ABBL05] Alferes, J.J., Banti, F., Brogi, A., Leite, J.A.: The refined extension principle for semantics of dynamic logic programming. *Studia Logica* 79(1), 7–32 (2005)
- [AGDO06] Acosta-Guadarrama, J.C., Dix, J., Osorio, M.: Update sequences in generalised answer set programming based on structural properties. In: Kellenberger, P. (ed.) *Special Session of the 5th International MICA Conference*, pp. 32–41. IEEE Computer Society Press, California (2006)
- [AGDOZ05] Acosta-Guadarrama, J.C., Dix, J., Osorio, M., Zacarías, F.: Updates in Answer Set Programming based on structural properties. In: McIlraith, S., Peppas, P., Thielscher, M. (eds.) *7th International Symposium on Logical Formalizations of Commonsense Reasoning*, Corfu, Greece, pp. 213–219. Fakultät Informatik (2005)
- [BG03] Balduccini, M., Gelfond, M.: Logic programs with consistency-restoring rules. In: *Proceedings of the AAAI Spring 2003 Symposium*, pp. 9–18. AAAI Press, California (2003)
- [Bre02] Brewka, G.: Logic programming with ordered disjunction. In: *AAAI 2002. Proceedings of the 18th National Conference on Artificial Intelligence*, Morgan Kaufmann, San Francisco (2002)

- [CDE<sup>+</sup>02] Calimeri, F., Dell'Armi, T., Eiter, T., Faber, W., Gottlob, G., Ianni, G., Ielpa, G., Koch, C., Leone, N., Perri, S., Pfeifer, G., Polleres, A.: The DLV System. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, Springer, Heidelberg (2002)
- [EFST02] Eiter, T., Fink, M., Sabbatini, G., Tompits, H.: On properties of update sequences based on causal rejection. *TPLP* 2(6), 711–767 (2002)
- [GL88] Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R.A., Bowen, K.A. (eds.) *ICLP/SLP. Logic Programming, Proceedings of the Fifth International Conference and Symposium*, MIT Press, Cambridge (1988)
- [GL91] Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9(3/4), 365–386 (1991)
- [KM90] Kakas, A.C., Mancarella, P.: Generalized Stable Models: A semantics for abduction. In: *ECAI, Stockholm, Sweden*, pp. 385–391 (1990). Aiello L.
- [KM91a] Katsuno, H., Mendelzon, A.O.: On the difference between updating a knowledge base and revising it. In: *KR 1991, Morgan Kaufmann Publishers, USA* (1991)
- [KM91b] Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. *Artificial Intelligence* 52(3), 263–294 (1991)
- [LPF<sup>+</sup>06] Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic* 7(3), 499–562 (2006)
- [NS97] Niemela, I., Simons, P.: Smodels — an implementation of the stable model and well-founded semantics for normal logic programs. In: Fuhrbach, U., Dix, J., Nerode, A. (eds.) *LPNMR 1997. LNCS*, vol. 1265, pp. 420–429. Springer, Heidelberg (1997)
- [SI03] Sakama, C., Inoue, K.: An abductive framework for computing knowledge base updates. *TPLP* 3(6), 671–715 (2003)
- [ZF05] Zhang, Y., Foo, N.: A unified framework for representing logic program updates. In: Veloso, M.M., Kambhampati, S. (eds.) *AAAI 2005. Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, USA, pp. 707–713. AAAI Press / The MIT Press (2005)

# On Reachability of Minimal Models of Multilattice-Based Logic Programs\*

Jesús Medina, Manuel Ojeda-Aciego, and Jorge Ruiz-Calviño

Dept. Matemática Aplicada. Universidad de Málaga  
{jmedina, aciego, jorgeruical}@ctima.uma.es

**Abstract.** In this paper some results are obtained regarding the existence and reachability of minimal fixed points for multiple-valued functions on a multilattice. The concept of inf-preserving multi-valued function is introduced, and shown to be a sufficient condition for the existence of minimal fixed point; then, we identify a sufficient condition granting that the immediate consequence operator for multilattice-based fuzzy logic programs is sup-preserving and, hence, computes minimal models in at most  $\omega$  iterations.

## 1 Introduction

Multilattice-based logic programs have been recently introduced as an extended paradigm for fuzzy logic programming in which the underlying set of truth-values for the propositional variable is considered to have a more relaxed structure than that of a complete lattice.

This line of research follows the trend of generalising the structure of the underlying set of truth-values for fuzzy logic programming, which has attracted the attention of a number of researchers in the recent years. For instance, there are approaches to fuzzy logic programming which are based either on the structure of lattice (residuated lattice [1,2] or multi-adjoint lattice [3]), or on more restrictive structures, such as bilattices [4,5], specially suited for the treatment of non-isotonicity, or even trilattices [6], in which points can be ordered according to truth, information, or precision. More general structures such as algebraic domains [7] have been used as well.

The first definition of multilattices seems to have been introduced in [8], although, much later, other authors proposed slightly different approaches [9,10], the later being more appealing to computation.

The crucial point in which a complete multilattice differs from a complete lattice is that a given subset does not necessarily has a least upper bound (resp. greatest lower bound) but some minimal (resp. maximal) ones. As far as we know, the first paper which used multilattices in the context of extended fuzzy logic programming was [11], which was later generalized in [13]. In these

---

\* Partially supported by Andalusian project P06-FQM-02049 and Spanish project TIN2006-15455-C03-01.

papers, the meaning of programs was defined by means of a fixed point semantics. In particular, the non-existence of suprema in general, but a set of minimal upper bounds, suggested the possibility of developing a non-deterministic fixed point theory in the form of a multi-valued immediate consequences operator. Essentially, the results presented were the existence of minimal models below any model of a program, and that any minimal model can be attained by the iteration of a suitable version of the immediate consequence operator, existence of minimal models was proved independently of the fixed-point semantics used to reach them; but some other problems remained open, such as the constructive nature of minimal models or the reachability of minimal models after at most countably many iterations.

The first contribution of this paper is a theoretical one, related to the existence of minimal fixed points: obviously, the main theoretical problem can be stated simply in terms of a suitable version of fixed point theorem for multi-valued functions on a multilattice. Here, we provide an existence result for minimal fixed-points in such a general context.

The second contribution relates to the reachability of minimal models; specifically, we introduce conditions guaranteeing that minimal models can be reached by a suitable iteration of the immediate consequences operator, the underlying idea is to give a general version of a related result presented in [13] but for single-valued functions.

The structure of the paper is as follows: in Section 2, the definition and some preliminary results about multilattices are presented; later, in Section 3, we move to the context of multi-valued functions and orbits on a multilattice, in order to set the basic results about the existence of fixed-point for such functions; then, in Section 4, we concentrate on the case of fuzzy logic programs evaluated on a multilattice, the main result being the introduction of sufficient conditions granting computability for its fixed-point semantics; finally, Section 5 concludes.

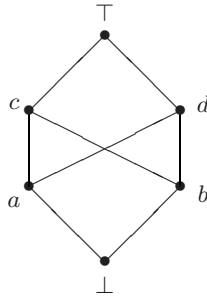
## 2 Preliminaries

We provide in this section the basic notions of the theory of multilattices, together with some preliminary results which will be used later in this paper.

**Definition 1.** *A complete multilattice is a partially ordered set,  $\langle M, \preceq \rangle$ , such that for every subset  $X \subseteq M$ , the set of upper (resp. lower) bounds of  $X$  has minimal (resp. maximal) elements, which are called multi-suprema (resp. multi-infima).*

The sets of multi-suprema and multi-infima of a set  $X$  are denoted by  $\text{multisup}(X)$  and  $\text{multinf}(X)$ . It is straightforward to note that these sets consist of pairwise incomparable elements (also called antichains).

*Example 1.* The simplest example of proper multilattice (i.e. one which is not a lattice) is called  $M6$  and is shown in Fig. 1.



**Fig. 1.** The multilattice  $M_6$

An arbitrary complete multilattice needs not have nice computational properties. As an example of counter-intuitive behaviour, simply note that an upper bound of a set  $X$  needs not be greater than any minimal upper bound (multi-supremum); such a condition (and its dual, concerning lower bounds and multi-infima) has to be explicitly required.

The fulfilment of this condition is called *coherence*, and is formally introduced in the following definition, where we use the Egli-Milner pre-ordering relation, i.e.,  $X \sqsubseteq_{EM} Y$  if and only if for every  $y \in Y$  there exists  $x \in X$  such that  $x \preceq y$  and for every  $x \in X$  there exists  $y \in Y$  such that  $x \preceq y$ .

**Definition 2.** A complete multilattice  $M$  is said to be coherent if the following pair of inequations hold for all  $X \subseteq M$ :

$$LB(X) \sqsubseteq_{EM} \text{multinf}(X); \quad \text{multisup}(X) \sqsubseteq_{EM} UB(X)$$

Coherence together with the non-existence of infinite antichains (so that the sets  $\text{multisup}(X)$  and  $\text{multinf}(X)$  are always finite) have been shown to be useful conditions when working with multilattices. Under these hypotheses, the following important result was obtained in [11]:

**Lemma 1.** Let  $M$  be a coherent complete multilattice without infinite antichains, then any chain in  $M$  has a supremum and an infimum.

Now that we have given the basic results for a multilattice, we turn our attention to preliminary results for functions defined on a multilattice.

The definitions of isotone and inflationary function are the standard ones also in the framework of multilattices. We recall these definitions below:

**Definition 3.** Let  $f: M \rightarrow M$  be a function on a multilattice, then:

- $f$  is isotone if and only if for every  $x, y \in M$  such that  $x \preceq y$  we have that  $f(x) \preceq f(y)$ .
- $f$  is inflationary if and only if  $x \preceq f(x)$  for every  $x \in M$

For isotone and inflationary functions on a multilattice we have the following result concerning fixed points, introduced in [13]:

**Theorem 1.** *Let  $M$  be a coherent complete multilattice without antichains, let  $f: M \rightarrow M$  be an isotone and inflationary mapping on a multilattice, then its set of fixed points is non-empty and has a minimum element.*

As stated in the introduction, the main theoretical problem in this paper is to extend the previous theorem to the framework of multiple-valued functions.

### 3 Multi-valued Functions and Orbits on Multilattices

In this section we will recall some important results included in [12] about how to reach minimal fixed points of multi-valued functions. But another important point is the existence of this minimal fixed points. Therefore, we will search sufficient conditions to ensure the existence of these points.

Firstly we need recall some preliminary definitions.

**Definition 4.** *Given a multilattice  $(M, \leq)$ , by a multi-valued function we mean a function  $f: M \rightarrow 2^M$  (we do not require that  $f(x) \neq \emptyset$  for every  $x \in M$ ).*

*We say that  $x \in M$  is a fixed point of  $f$  if and only if  $x \in f(x)$ .*

Although there exist different definitions of orders in  $2^M$ , we will consider in this paper just the Smyth pre-ordering among sets, and we will write  $X \sqsubseteq_S Y$  if and only if for every  $y \in Y$  there exists  $x \in X$  such that  $x \leq y$ . This pre-order is used to define the isotonicity and inflation for multi-valued functions.

**Definition 5.** *Given a multilattice  $(M, \leq)$ , a multi-valued function  $f: M \rightarrow 2^M$  it is called:*

- Isotone if and only if  $x \leq y$  implies  $f(x) \sqsubseteq_S f(y)$ , for all  $x, y \in M$ .
- Inflationary if and only if  $\{x\} \sqsubseteq_S f(x)$  for every  $x \in M$ .

The concept of orbit has proven to be an important tool for studying reachability of minimal fixed points, see [18].

**Definition 6.** *Let  $f: M \rightarrow 2^M$  be a multi-valued function, an orbit of  $f$  is a transfinite sequence  $(x_i)_{i \in I}$  of elements  $x_i \in M$  where the cardinality of  $M$  is less than the cardinality of  $I$  ( $|M| < |I|$ ) and:*

$$\begin{aligned} x_0 &= \perp \\ x_{i+1} &\in f(x_i) \\ x_\alpha &\in \text{multisup}\{x_i \mid i < \alpha\}, \text{ for limit ordinals } \alpha \end{aligned}$$

As  $f(x_i)$  is a set we have many possible choices for  $x_{i+1}$  so we have many possible orbits. Note the following straightforward consequences of the definition:

1. In an orbit, we have  $f(x_i) \neq \emptyset$  for every  $i \in I$ .
2. If  $(x_i)_{i \in I}$  is an orbit of  $f$  and there exists  $k \in I$  such that  $x_k = x_{k+1}$ , then  $x_k$  is a fixed point of  $f$ .

3. Any increasing orbit eventually reaches a fixed point (this follows from the inequality  $|M| < |I|$ ).

From the third point above, if we can show the existence of such orbits, then we ensure the existence of fixed points.

To begin with, if  $f$  is inflationary, any orbit  $(x_i)_{i \in I}$  is increasing, for successor ordinals the inequality  $\{x_i\} \sqsubseteq_S f(x_i)$  follows by inflation, hence  $x_i \preceq x_{i+1}$ . The definition for limit ordinals, directly implies that it is greater than any of its predecessors.

Furthermore, any orbit converges to a fixed point of  $f$ . This follows directly, since every transfinite increasing sequence is eventually stationary, and an ordinal  $\alpha$  such that  $x_\alpha = x_{\alpha+1} \in f(x_\alpha)$  is a fixed point.

Propositions 1 and 2 below were introduced in [12] and show conditions under which any minimal fixed point is attained by means of an orbit:

**Proposition 1.** *For an inflationary and isotone multi-valued function  $f$  we have that: for any minimal fixed point there is an orbit converging to it.*

**Proposition 2.** *If a multi-valued function  $f$  is inflationary, isotone and sup-preserving, then at most countably many steps are necessary to reach a minimal fixed point (provided that some exists).*

In order to find conditions to the existence of minimal fixed points of multi-valued functions we will follow the usual practice of considering the sets of pre-fixed points and post-fixed points:

$$\begin{aligned} \Phi(f) &= \{x \in M \mid f(x) \sqsubseteq_S \{x\}\} \\ \Psi(f) &= \{x \in M \mid \{x\} \sqsubseteq_S f(x)\} \end{aligned}$$

Note that,  $\Psi(f)$  is always nonempty, since  $\perp \in \Psi(f)$ . However, this does not hold for  $\Phi(f)$ , since it is not always the case that  $\top \in \Phi(f)$ . Actually, it is easy to see that  $\top \in \Phi(f)$  if and only if  $f(\top) \neq \emptyset$  (but recall that  $f(\top)$  can be empty). For a general element  $x$ , the previous equivalence does not hold, but only one implication: if  $f(x) = \emptyset$  then  $x \notin \Phi(f)$  so if  $x \in \Phi(f)$  then  $f(x) \neq \emptyset$ .

The following general result gives us a characterisation of the fixed point of a multi-valued function in terms of  $\Phi(f)$ :

**Proposition 3.** *Let  $M$  be a multilattice and  $f: M \rightarrow 2^M$  an inflationary multi-valued function. Then  $x \in \Phi(f)$  if and only if  $x$  is a fixed point of  $f$ .*

*Proof.* Let  $x \in \Phi(f)$  then, as  $f$  is inflationary, we have that  $\{x\} \sqsubseteq_S f(x) \sqsubseteq_S \{x\}$ . Hence for  $x \in \{x\}$  we have that there exists  $y \in f(x)$  such that  $x \preceq y \preceq x$ , so  $x = y \in f(x)$ . The other implication holds trivially.  $\square$

It is not difficult to find examples which show that the inflationary requirement is essential.

Note that by the proposition above the existence of minimal fixed points of an inflationary multi-valued function is equivalent to the existence of minimal

elements of  $\Phi(f)$ , therefore we will look for conditions to ensure the existence of minimal fixed points of  $\Phi(f)$ .

The previous proposition also holds when  $f$  is isotone, as a result under either isotone or inflationary  $f$  all the minimal elements of  $\Phi(f)$  are minimal fixed points. This is established in the following theorem.

**Theorem 2.** *Let  $f: M \rightarrow 2^M$  be a isotone or inflationary multi-valued function. If  $\Phi(f)$  has minimal elements then these minimal elements are minimal fixed points of  $f$ .*

*Proof.* Let  $f$  be isotone, and  $y$  a minimal element of  $\Phi(f)$ , so  $\emptyset \neq f(y) \sqsubseteq_S \{y\}$  and there exists  $y' \in f(y)$  such that  $y' \preceq y$ .

As  $f$  is isotone we have that  $f(y') \sqsubseteq_S f(y)$ , hence, since  $y' \in f(y)$  there exists  $y'' \in f(y')$  with  $y'' \preceq y'$ . Therefore,  $f(y') \sqsubseteq_S \{y'\}$  and  $y' \in \Phi(f)$  and  $y' \preceq y$  but  $y$  is minimal in  $\Phi(f)$ , so  $y = y' \in f(y)$  and  $y$  is a fixed point of  $f$ .

Let us see now that  $y$  is a minimal fixed point. Assume that  $x$  is a fixed point of  $f$  with  $x \preceq y$ . As,  $x$  is a fixed point we have that  $x \in \Phi(f)$  but then, by the minimality of  $y$  in  $\Phi(f)$ , we would have  $x = y$  and  $y$  is a minimal fixed point as well.

The case of  $f$  inflationary follows easily from Proposition 3. □

In order to give some conditions to ensure the existence of minimal elements of  $\Phi(f)$ , for multi-valued function on a multilattice, we will consider some kind of ‘continuity’ in our multi-valued functions. This continuity is understood in the sense of preservation of suprema and infima; but, obviously, we have to state formally what this preservation is meant since in complete multilattices we only have for granted the existence of *sets of multi-infima and sets of multi-suprema*.

In this context, it is convenient again to rely on coherent complete multilattices  $M$  without infinite antichains so that, at least, we have the existence of suprema and infima of chains.

**Definition 7.** *A multi-valued function  $f: M \rightarrow 2^M$  is said to be sup-preserving if and only if for every chain<sup>1</sup>  $X = (x_i)_{i \in I}$  we have that:*

$$f(\text{sup}\{x_i \mid i \in I\}) = \{y \mid \text{there are } y_i \in f(x_i) \text{ s.t. } y \in \text{multisup}\{y_i \mid i \in I\}\}$$

*A multi-valued function  $f: M \rightarrow 2^M$  is inf-preserving if and only if for every chain  $X = (x_i)_{i \in I}$  we have that:*

$$f(\text{inf}\{x_i \mid i \in I\}) = \{y \mid \text{there are } y_i \in f(x_i) \text{ s.t. } y \in \text{multinf}\{y_i \mid i \in I\}\}$$

The following theorem states that the property of being inf-preserving is a sufficient condition to ensure the existence of minimal fixed points of a multi-valued function.

**Theorem 3.** *Let  $f: M \rightarrow 2^M$  be an inf-preserving multi-valued function with  $\Phi(f) \neq \emptyset$ , then  $\Phi(f)$  has minimal elements.*

---

<sup>1</sup> A chain  $X$  is a totally ordered subset and, for convenience, will be denoted as an indexed set  $(x_i)_{i \in I}$ .



*Proof.* We will apply Zorn’s lemma, and prove that every chain of elements of  $\Phi(f)$  has infimum in  $\Phi(f)$ .

By hypothesis  $\Phi(f) \neq \emptyset$ . Let  $(x_i)_{i \in I}$  be a chain of elements of  $\Phi(f)$  and consider  $x = \inf\{x_i \mid i \in I\}$  (which exists by Lemma [1](#)).

In order to prove that  $x \in \Phi(f)$ , we will prove the existence of a particular element of  $f(x)$  which is smaller than  $x$ .

Firstly, as  $x_i \in \Phi(f)$  we have that for all  $i \in I$  there exists  $y_i \in f(x_i)$  such that  $y_i \preceq x_i$ . Now, consider an element  $y \in \text{multinf}\{y_i \mid i \in I\}$ . It is straightforward to note that the inequality  $y \preceq \inf\{x_i \mid i \in I\} = x$  holds. Now, as  $f$  is inf-preserving, we know that

$$f(x) = \{z \mid \text{there are } y_i \in f(x_i) \text{ s.t. } z \in \text{multinf}\{y_i \mid i \in I\}\}$$

hence, we have  $y \in f(x)$  and, consequently  $f(x) \sqsubseteq_S \{x\}$ . We have proved that  $\Phi(f)$  is closed for the infima of chains and, in particular, by Zorn’s lemma,  $\Phi(f)$  has minimal elements.  $\square$

Note that it is easy to check that an inf-preserving function is isotone, thus by a combination of Theorems [2](#) and [3](#) we obtain the existence of minimal fixed points of  $f$ .

It is worth to note that this result does not apply directly to the context of fuzzy logic programs on a multilattice, since minimal fixed-points are known to exist under the only assumptions of coherence and absence of infinite antichains of the underlying multilattice.

However, the obtained result follows the line of several versions of fixed point theorems for multi-valued functions on a lattice already present in the literature [\[14,15,16,17,18\]](#). It is remarkable that most of these results were developed to be used in the context of the study of Nash equilibria of supermodular games, but extending the study in this direction is out of the scope of this paper.

## 4 On Fuzzy Logic Programs on a Multilattice

The previous results will be applied to the particular case of the immediate consequences operator for logic programs on a multilattice, as defined in [\[13\]](#).

To begin with we will recall the definition of the fuzzy logic programs based on a multilattice:

**Definition 8.** A logic program based on a multilattice  $(M, \preceq)$  is a set  $\mathbb{P}$  of rules of the form  $A \leftarrow \mathcal{B}$  such that:

- $A$  is a propositional symbol, and
- $\mathcal{B}$  is a formula built from propositional symbols and elements of  $M$  by using isotone operators.

In general, non-atomic formulae will be represented by  $\mathcal{B} = @ (B_1, \dots, B_n)$  where  $@$  denotes the composition of the isotone operators involved in the construction of  $\mathcal{B}$ , and  $B_i$  are either propositional symbols or elements of  $M$ .

The definition of interpretation and model of a program is given as follows:

**Definition 9**

- An interpretation is a mapping  $I$  from the set of propositional symbols to  $M$ .
- We say that  $I$  satisfies a rule  $A \leftarrow B$  if and only if  $\hat{I}(B) \preceq I(A)$ , where  $\hat{I}$  is the homomorphic extension<sup>2</sup> of  $I$  to the set of all formulae.
- An interpretation  $I$  is said to be a model of a program  $\mathbb{P}$  iff all rules in  $\mathbb{P}$  are satisfied by  $I$ .

A fixed point semantics was given by means of the following consequences operator.

**Definition 10.** Consider a fuzzy logic program  $\mathbb{P}$  based on a multilattice, an interpretation  $I$ , and a propositional symbol  $A$ ; the immediate consequences operator is defined as follows:

$$T_{\mathbb{P}}(I)(A) = \text{multisup} \left( \{I(A)\} \cup \{\hat{I}(B) \mid A \leftarrow B \in \mathbb{P}\} \right)$$

It is easy to see by the very definition that the immediate consequences operator is an inflationary multi-valued function defined on the set of interpretation of the program  $\mathbb{P}$ , which is a multilattice. Moreover, models of a program  $\mathbb{P}$  are characterized as follows.

**Proposition 4 (see [11]).** An interpretation  $I$  is a model of a program if and only if  $I(A) \in T_{\mathbb{P}}(I)(A)$  for all propositional symbol  $A$ .

The requirement that  $M$  is a coherent multilattice without infinite antichains was imposed in [11] in order to prove the existence of minimal fixed points. Then, a straightforward application of Proposition 2 generated the following result:

**Theorem 4 (see [12]).** If  $T_{\mathbb{P}}$  is sup-preserving, then  $\omega$  steps are sufficient to reach a minimal model.

In the rest of the section, we will concentrate on the condition of  $T_{\mathbb{P}}$  being sup-preserving. To begin with, let us show that some part of the condition is always fulfilled:

**Lemma 2.** Let  $\{I_i\}_{i \in \Lambda}$  be a chain, then the following inequality holds

$$\left\{ J \mid \text{there are } J_i \in T_{\mathbb{P}}(I_i) \text{ with } J \in \text{multisup}_{i \in \Lambda} \{J_i\} \right\} \sqsubseteq_S T_{\mathbb{P}}(\text{sup}_{i \in \Lambda} \{I_i\})$$

*Proof.* Consider  $I = \text{sup}_{i \in \Lambda} \{I_i\}$  and  $K \in T_{\mathbb{P}}(I)$ , we have that  $I_i \preceq I$  for every  $i \in \Lambda$ ; as  $T_{\mathbb{P}}$  is isotone we have that  $T_{\mathbb{P}}(I_i) \sqsubseteq_S T_{\mathbb{P}}(I)$  then for this  $K$  there are  $J_i \in T_{\mathbb{P}}(I_i)$  such that  $J_i \preceq K$  for every  $i \in \Lambda$ . Therefore,  $K \in \text{UB}\{J_i\}$  hence by coherence we have that there is  $L \in \text{multisup}\{J_i\}$  such that  $L \preceq K$  and by construction  $L \in \{J \mid \text{there are } J_i \in T_{\mathbb{P}}(I_i) \text{ with } J \in \text{multisup}\{J_i\}\}$   $\square$

<sup>2</sup> The homomorphic extension  $\hat{I}$  of  $I$  applied to a non-atomic formula  $@(B_1, \dots, B_n)$ , is defined as follows:  $\hat{I}(@ (B_1, \dots, B_n)) = @(I(B_1), \dots, I(B_n))$ .

If we want to get the other inequality we need to assume that  $T_{\mathbb{P}}(I)(A)$  is a singleton for all  $I$  and  $A$  (which we will call that  $T_{\mathbb{P}}$  is a singleton) as the next theorem shows:

**Theorem 5.** *If  $T_{\mathbb{P}}$  is a singleton for every interpretation  $I$  and the operators of the body of  $\mathbb{P}$  are sup-preserving<sup>3</sup>, then  $T_{\mathbb{P}}$  is sup-preserving.*

*Proof.* We have to prove that for every chain of interpretations  $\{I_i\}_{i \in \Lambda}$  we have that

$$T_{\mathbb{P}}(\sup_{i \in \Lambda} \{I_i\}) = \{J \mid \text{there are } J_i \in T_{\mathbb{P}}(I_i) \text{ with } J \in \text{multisup}_{i \in \Lambda} \{J_i\}\} \quad (1)$$

First of all, as  $T_{\mathbb{P}}$  is a singleton for every interpretation we have that the left hand side of equality (1) is a singleton, so we have to see that the right part is a singleton too. By hypothesis, we have that  $T_{\mathbb{P}}(I_i)$  is a singleton, so there is only one possible choice of  $J_i$ . Moreover,  $T_{\mathbb{P}}$  is Smyth isotone, this, together with that  $T_{\mathbb{P}}(I_i)$  are singletons lead us to  $\{T_{\mathbb{P}}(I_i)\}_{i \in \Lambda}$  being a chain, so it has a supremum, namely  $J$ , and the right part of (1) is also a singleton. Therefore we have to prove that

$$T_{\mathbb{P}}(\sup_{i \in \Lambda} \{I_i\}) = \{J\}$$

Given  $I = \sup_{i \in \Lambda} \{I_i\}$ , by Lemma 2, we have that  $\{J\} \subseteq T_{\mathbb{P}}(I)$  since both are singletons. To prove the other inequality we will prove that for every propositional symbol,  $A$ , we have that the element in  $T_{\mathbb{P}}(I)(A)$  is less than or equal to  $J(A)$ , that  $T_{\mathbb{P}}(I)(A) \preceq J(A)$ .

By definition we have that<sup>4</sup>

$$T_{\mathbb{P}}(I)(A) = \sup\{I(A) \cup \{\hat{I}(\mathcal{B}) \text{ with } A \leftarrow \mathcal{B} \in \mathbb{P}\}\}$$

and we have that  $J(A) = \sup\{J_i(A)\}$ , where

$$J_i(A) = T_{\mathbb{P}}(I_i)(A) = \sup\{I_i(A) \cup \{I_i(\mathcal{B}) \text{ with } A \leftarrow \mathcal{B} \in \mathbb{P}\}\}$$

so  $I_i(A) \preceq J(A)$  for every  $i \in \Lambda$  and therefore  $I(A) \preceq J(A)$

Now we will see that  $I(\mathcal{B}) \preceq J(A)$  for every  $A \leftarrow \mathcal{B} \in \mathbb{P}$ . If  $\mathcal{B}$  is a fact or is a propositional symbol then the inequality is trivial. Let us suppose that  $\mathcal{B}$  is of the form  $@[B_1, B_2]$ , the case of  $n$  propositional symbols is proved in a similar way. We have that,  $I_i(\mathcal{B}) \preceq J(A)$  for every  $i \in \Lambda$ , so  $@[I_i(B_1), I_i(B_2)] \preceq J(A)$  for every  $i \in \Lambda$ , therefore we have that  $\sup_{i \in \Lambda} \{@[I_i(B_1), I_i(B_2)]\} \preceq J(A)$ . On the other hand, we have that:  $\hat{I}(\mathcal{B}) = @[I(B_1), I(B_2)]$ .

As  $@$  is sup-preserving by hypothesis we have that:

$$\begin{aligned} @[I(B_1), I(B_2)] &= \sup_{j \in \Lambda} \{@[I_j(B_1), \sup_{l \in \Lambda} \{I_l\}(B_2)]\} \\ &= \sup_{j \in \Lambda} \{\sup_{l \in \Lambda} \{@[I_j(B_1), I_l(B_2)]\}\} \end{aligned}$$

<sup>3</sup> This is Definition 7 for single-valued functions. That is  $f(\sup_{i \in \Lambda} \{x_i\}) = \sup_{i \in \Lambda} \{f(x_i)\}$  for all chain  $(x_i)_{i \in \Lambda}$ .

<sup>4</sup> In the definition of  $T_{\mathbb{P}}$  it is multisup instead of sup but as  $T_{\mathbb{P}}$  is a singleton for all  $I$  and  $A$  we will write, abusing the notation, sup.

Now, as  $\{I_i\}_{i \in A}$  is a chain we can suppose that  $I_l \preceq I_j$ , hence

$$\begin{aligned} \sup_{j \in A} \{ \sup_{l \in A} \{ \textcircled{[} I_j(B_1), I_l(B_2) \textcircled{]} \} \} &\preceq \sup_{j \in A} \{ \sup_{l \in A} \{ \textcircled{[} I_j(B_1), I_j(B_2) \textcircled{]} \} \} \\ &= \sup_{j \in A} \{ \textcircled{[} I_j(B_1), I_j(B_2) \textcircled{]} \} \end{aligned}$$

and we have that  $\hat{I}(\mathcal{B}) \preceq J(A)$ , so we have proved that for every  $A$ :

$$T_{\mathbb{P}}(I)(A) = \sup \{ I(A) \cup \{ \hat{I}(\mathcal{B}) \text{ with } A \leftarrow \mathcal{B} \in \mathbb{P} \} \} \preceq J(A) \quad \square$$

*Remark 1.* Both conditions are necessary in the Theorem, as we can see in the following examples:

- Let us consider in  $M6$  the program  $A \leftarrow B$  (there is no conjunctors so they are sup-preserving), and the interpretations  $I_1(A) = b$ ,  $I_1(B) = a$  and  $I_2(A) = c$ ,  $I_2(B) = a$  we have that  $I_1 \preceq I_2$  and that  $T_{\mathbb{P}}(I_1)(A) = \{c, d\}$  ( $T_{\mathbb{P}}$  is not a singleton) and  $T_{\mathbb{P}}(I_2)(A) = \{c\}$ .

If  $T_{\mathbb{P}}$  is sup-preserving, then we would have that

$$\begin{aligned} T_{\mathbb{P}}(I_2)(B) &= T_{\mathbb{P}}(\sup \{ I_1, I_2 \})(B) \\ &= \{ y \mid \text{there are } y_i \in T_{\mathbb{P}}(I_i)(B) \text{ with } y \in \text{multisup} \{ y_1, y_2 \} \} \end{aligned}$$

but  $T_{\mathbb{P}}(I_2)(B) = \{c\}$  while the right part of the equality is  $\{c, \top\}$ .

- In the multilattice of Figure 2 let us consider the program with only one rule  $A \leftarrow B * C$ , where  $*$  is commutative and defined as follows:

$$\begin{aligned} x * x &= x; & \perp * x &= \perp; & \top * x &= \top \text{ if } x \neq \perp; & c_i * c_j &= c_{\min(i,j)} \\ c * c_i &= c_i; & c * d &= \top; & d * x &= x \text{ if } x \neq c \end{aligned}$$

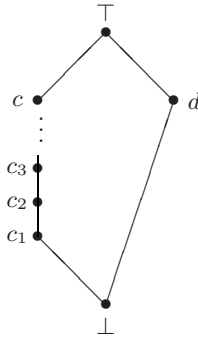


Fig. 2.

where  $x$  is an element of the multilattice of Figure 2. We have that  $*$  is not sup-preserving because:  $\sup \{c_i\} * d = c * d = \top \neq c = \sup \{c_i\} = \sup \{c_i * d\}$ . However, it is easy to see that  $T_{\mathbb{P}}$  is a singleton (we are in a lattice so multisup turns out to be sup). Now, if we consider the interpretations  $\{I_i\}_{i \in \mathbb{N}}$  defined as  $I_i(A) = c$ ;  $I_i(B) = c_i$ ;  $I_i(C) = d$  we have that  $\{I_i\}_{i \in \mathbb{N}}$  is a chain whose supremum is the interpretation  $I$  defined as  $I(A) = c$ ;  $I(B) = c$ ;  $I(C) = d$ .

If  $T_{\mathbb{P}}$  were sup-preserving then we would have that:

$$\begin{aligned} T_{\mathbb{P}}(I)(A) &= T_{\mathbb{P}}(\sup \{I_i\}_{i \in \mathbb{N}})(A) \\ &= \{y \mid \text{there are } y_i \in T_{\mathbb{P}}(I_i)(A) \text{ with } y \in \text{multisup}\{y_i\}_{i \in \mathbb{N}}\} \end{aligned}$$

but  $T_{\mathbb{P}}(I)(A) = \{\top\}$  while  $T_{\mathbb{P}}(I_i)(A) = \{c\}$  for every  $i \in \mathbb{N}$  so the right part of the equality is  $\{c\}$ . Thus,  $T_{\mathbb{P}}$  is a singleton for every interpretation but not sup-preserving.

## 5 Conclusions

We have presented a prospective study of the theory of fixed points of multiple-valued functions defined on a multilattice, continuing the study of computational properties of multilattices initiated in [11,13].

Some general results have been obtained regarding the existence of minimal fixed points for multiple-valued functions as well as their reachability in at most countably many steps by means of an iterative procedure finishing with the presentation of some conditions which ensure the hypotheses for that reachability in countably many steps.

As an application of this theoretical result, we have shown that when the immediate consequences operator of a fuzzy logic program is sup-preserving (in the sense formally given above), then there is no need of transfinite iteration in order to attain any minimal fixed point.

The starting point has been to consider the different versions of fixed point theorems for multi-valued functions on a lattice already present in the literature [14,15,16,17,18]. It is remarkable that most of these results were developed to be used in the context of the study of Nash equilibria of supermodular games.

## References

1. Damásio, C., Pereira, L.M.: Monotonic and residuated logic programs. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 748–759. Springer, Heidelberg (2001)
2. Vojtáš, P.: Fuzzy logic programming. Fuzzy sets and systems 124(3), 361–370 (2001)
3. Medina, J., Ojeda-Aciego, M.: Multi-adjoint logic programming with continuous semantics. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) LPNMR 2001. LNCS (LNAI), vol. 2173, pp. 351–364. Springer, Heidelberg (2001)
4. Fitting, M.: Bilattices and the semantics of logic programming. Journal of Logic Programming 11, 91–116 (1991)
5. Loyer, Y., Straccia, U.: Epistemic foundation of the well-founded semantics over bilattices. In: Fiala, J., Koubek, V., Kratochvíl, J. (eds.) MFCS 2004. LNCS, vol. 3153, pp. 513–524. Springer, Heidelberg (2004)
6. Lakshmanan, L.V.S., Sadri, F.: On a theory of probabilistic deductive databases. Theory and Practice of Logic Programming 1(1), 5–42 (2001)
7. Rounds, W., Zhang, G.-Q.: Clausal logic and logic programming in algebraic domains. Information and Computation 171, 183–200 (2001)

8. Benado, M.: Les ensembles partiellement ordonnés et le théorème de raffinement de Schreier, II. Théorie des multistruktures. *Czechoslovak Mathematical Journal* 5(80), 308–344 (1955)
9. Hansen, D.: An axiomatic characterization of multilattices. *Discrete Mathematics* 1, 99–101 (1981)
10. Martínez, J., Gutiérrez, G., de Guzmán, I., Cordero, P.: Generalizations of lattices via non-deterministic operators. *Discrete Mathematics* 295, 107–141 (2005)
11. Medina, J., Ojeda-Aciego, M., Ruiz-Calviño, J.: Multi-lattices as a basis for generalized fuzzy logic programming. In: Bloch, I., Petrosino, A., Tettamanzi, A.G.B. (eds.) *WILF 2005. LNCS (LNAI)*, vol. 3849, pp. 61–70. Springer, Heidelberg (2006)
12. Medina, J., Ojeda-Aciego, M., Ruiz-Calviño, J.: A fixed-point theorem for multi-valued functions with application to multilattice-based logic programming. *Lect. Notes in Computer Science*, vol. 4578, pp. 37–44 (2007)
13. Medina, J., Ojeda-Aciego, M., Ruiz-Calviño, J.: Fuzzy logic programming via multilattices. *Fuzzy Sets and Systems* 158(6), 674–688 (2007)
14. d’Orey, V.: Fixed point theorems for correspondences with values in a partially ordered set and extended supermodular games. *Journal of Mathematical Economics* 25, 345–354 (1996)
15. Echenique, F.: A short and constructive proof of Tarski’s fixed-point theorem. *International Journal of Game Theory* 33, 215–218 (2005)
16. Stouti, A.: A generalized Amman’s fixed point theorem and its application to Nash equilibrium. *Acta Mathematica Academiae Paedagogicae Nyíregyháziensis* 21, 107–112 (2005)
17. Zhou, L.: The set of Nash equilibria of a supermodular game is a complete lattice. *Games and economic behavior* 7, 295–300 (1994)
18. Khamsi, M.A., Misane, D.: Fixed point theorems in logic programming. *Annals of Mathematics and Artificial Intelligence* 21, 231–243 (1997)

# Update Sequences Based on Minimal Generalized Pstable Models

Mauricio Osorio<sup>1</sup> and Claudia Zepeda<sup>2</sup>

<sup>1</sup> Universidad de las Américas, CENTIA,  
Sta. Catarina Mártir, Cholula, Puebla, 72820 México  
osoriomauri@gmail.com

<sup>2</sup> Universidad Politécnica de Puebla,  
Tercer Carril del Ejido Serrano, San Mateo Cuanala,  
Municipio Juan C. Bonilla, Puebla, 72640 México  
czepedac@gmail.com

**Abstract.** In case intelligent agents get new knowledge and this knowledge must be added or updated to their knowledge base, it is important to avoid inconsistencies. Currently there are several approaches dealing with updates. In this paper, we propose a semantics for update sequences. We start introducing the notion of minimal generalized (MG) pstable models that, as we argue is interesting by itself. Based on MG pstable models we construct our update semantics. In this work, we also use some representative examples to compare our update semantics to other known update semantics and observe some advantages of it.

**Keywords:** Logic Programming, Update sequences.

## 1 Introduction

When intelligent agents get new knowledge and this knowledge must be added or updated to their knowledge base, then it is important to use an approach to deal with updates in order to avoid inconsistencies. For instance, if we want to consider the update of program  $Q_1 = \{-a, b\}$  with program  $Q_2 = \{a\}$  then it is evident that the simple union of these two programs generates an inconsistency. So it is necessary to apply some update approach to update  $Q_1$  with  $Q_2$ . It is easy to see that that the result of this update must be  $\{a, b\}$  since, the update approaches consider that new information has higher priority than the previous one, so  $Q_2$  has higher priority than  $Q_1$ .

Currently there are several approaches in non-monotonic reasoning dealing with updates, such as [5,9,4,12]. The update approach in [5] proposes to incorporate new information into the current knowledge base depending on a causal rejection principle. A complete analysis about the properties that an update operator should have is also presented in [5]. An alternative but equivalent definition of update sequences as presented in [5] is presented in [9]. Many of the properties presented in [5] and some novel properties are analyzed in [9]. An interesting framework for update logic programs under the answer set semantics

that makes use of defaultification and preference is introduced in [4]. In [12], new information is incorporated into the current knowledge base subject to a concept called minimal extended generalized answer sets.

It is important to point out that most of the update approaches in non-monotonic reasoning are based on answer sets semantics [6], so their results do not agree with a classical logic point of view. We can make this clear with the following example. Let  $P_1 = \{a \leftarrow \neg b, a \leftarrow b\}$  and  $P_2 = \{b \leftarrow a\}$ . From a classical logic point of view, we would expect that  $\{a, b\}$  corresponds to the result of updating  $P_1$  with  $P_2$ . However, there is no a result when we apply the approach in [5] to update  $P_1$  with  $P_2$ , and  $\{\}$  is the result when we apply the approach in [12] to update  $P_1$  with  $P_2$ . This last result means that  $a$  and  $b$  are assumed to be false. Moreover, since classical logic identifies a class of formal logics that have been most intensively studied and most widely used, it turns out to be very useful to have some updates approaches that allow us to keep a compromise with such logic.

Hence, as part of the contribution of this paper, we propose an update semantics that allow us to keep a compromise with classical logic. So, using the update semantics proposed in this paper, the result of updating program  $P_1$  with program  $P_2$  will be  $\{a, b\}$ . It is worth mentioning that the definition of the update semantics proposed in this paper is based on pstable semantics. At the same time, pstable semantics is based on  $G'_3$  which is a paraconsistent logic that has recently been studied in some detail in [8,10,11,3]. In particular, one of the results presented in [3] is that if an atom  $a$  is a logical consequence in classical logic of a program  $P$ , then  $P$  and the program  $P \cup \{a\}$  have the same pstable models, and even more, for any program  $Q$ ,  $P \cup Q$  and  $P \cup \{a\} \cup Q$  have the same pstable models.

Our semantics is based on a concept called minimal extended generalized pstable models. The definition of minimal extended generalized pstable models was inspired by a concept called minimal generalized (MG) answer sets [7,2]. We want to mention that currently MG answer sets have several applications: In [2] they were used to restore consistency, in [17] they were used as an alternative form to obtain the preferred plans of planning problems, in [17] they were also used as a simple way to get the preferred extensions of an argument framework, and in [16] they were used to define an update semantics for pairs of programs. Hence, we consider that the concept of minimal extended generalized pstable models can also have similar applications and it can be an alternative to those application using MG answer sets but considering a classical logic point of view.

In this paper we also present a second update semantics. It represents an alternative semantics to the first one and it can be also considered as another application of minimal extended generalized pstable models.

Our paper is structured as follows. In section [2] we introduce the preliminaries about general syntax of the logic programs and the pstable semantics based on  $G'_3$  logic. In section [3] we define the notion of minimal generalized pstable models. In section [4] we present an application of minimal extended pstable models: the semantics for updates consisting of a sequence of programs. In section [5] we



present an alternative to the semantics for update sequences given in section 4. Finally, in section 6 we present some conclusions and future work.

## 2 Preliminaries

In this paper, logic programs are understood as propositional theories. We shall use the language of propositional logic in the usual way, using propositional symbols:  $p, q, \dots$ , propositional connectives  $\wedge, \vee, \rightarrow, \neg, -$ ,  $\perp$  and auxiliary symbols:  $(, )$ . The unary connective  $-$  is restricted to appear only applied to propositional symbols. A formula is constructed as usual using propositional symbols and propositional connectives, but respecting the restriction of the connective  $-$  just mentioned. Note that we consider two types of negation: true or explicit negation (written as  $-$ ) and negation-as-failure (written as  $\neg$ ). An *atom* is a propositional symbol. A *literal* is either an atom  $a$  or the explicit negation of an atom  $-a$ . For a set  $S$  of literals, we define  $\neg S = \{\neg l \mid l \in S\}$ . We denote by  $Lit_{\mathcal{A}}$  the set  $\mathcal{A} \cup -\mathcal{A}$  of all literals over a set of atoms  $\mathcal{A}$ . For any well formed propositional formula  $f$ , we define a constraint as the formula  $f \rightarrow \perp$ . Constraints are considered as a special type of formula. A *regular theory* or *logic program* is just a finite set of well formed formulas or rules, it can be called just *theory* or *program* where no ambiguity arises. We shall define as a *rule* any well formed formula of the form:  $f \leftarrow g$ . A *basic program* is a regular theory that does not include neither true negation nor constraints. The signature of a logic program  $P$ , denoted as  $\mathcal{L}_P$ , is the set of atoms that occur in  $P$ . Sometimes we may use *not* instead of  $\neg$  and  $a, b$  instead of  $a \wedge b$ , following the traditional notation of logic programming. We want to stress the fact that in our approach, a logic program is interpreted as a propositional theory. We will restrict our discussion to propositional programs. We take for granted that programs with predicate symbols are only an abbreviation of the ground program.

We also clarify that by a consistent set of literals  $M$ , we mean that for all literals  $l \in M$  we have either  $l \in M$  or  $\neg l \in M$ , but not both. Finally, we define the complement of  $M$  as  $\widetilde{M} = Lit_{\mathcal{A}} \setminus M$ .

### 2.1 Pstable Semantics

We first consider our logic of interest that is called  $G'_3$ . This logic does not include the true negation connective.  $G'_3$  logic is defined through a 3-valued logic with truth values in the domain  $D = \{0, 1, 2\}$  where 2 is the designated value. The evaluation function of the logic connectives is then defined as follows:  $x \wedge y = \min(x, y)$ ;  $x \vee y = \max(x, y)$ ; and the  $\neg$  and  $\rightarrow$  connectives are defined according to the truth tables given in Table 1. Given a formula  $\alpha$ , we say that  $\alpha$  is a tautology if  $\alpha$  evaluates to the designated value for every valuation. For instance, we can verify that the formula  $a \rightarrow \neg\neg a$  is not a tautology since it evaluates to 0 when  $a$  is 1 and 0 is not the designated value. We also can verify that  $\neg\neg a \rightarrow a$  is a tautology since it evaluates to 2 when  $a$  is 0, 1 and 2. Given a basic program  $P$  and a formula  $\alpha$ , we write  $P \models \alpha$  if  $\bigwedge P \rightarrow \alpha$  is a tautology. For a given set of atoms  $M$  and a basic program  $P$  we say that  $M$  models  $P$  if

every formula in  $P$  evaluates to the designated value w.r.t to the evaluation that assigns the value 2 to every element in  $M$  and 0 otherwise.  $G'_3$  is a paraconsistent logic that has recently been studied in some detail in [8,10,11].

**Table 1.** Truth tables of connectives in  $G'_3$

$x$	$\neg x$	$\rightarrow$	0 1 2
0	2	0	2 2 2
1	2	1	0 2 2
2	0	2	0 1 2

Now we present the definition of the *pstable semantics* for basic programs. We call the construct  $P \cup \neg\widetilde{M}$  a *weak completion* of the program  $P$  (with respect to the set of atoms  $M$ ).

**Definition 1.** [11] Let  $P$  be any theory and let  $M$  be a set of atoms.  $M$  is a *pstable model* <sup>1</sup> of  $P$  if  $P \cup \neg\widetilde{M} \models M$  and  $M$  models  $P$ .

*Example 1.* Consider the following logic program:  $P = \{b \leftarrow \neg a, a \leftarrow \neg b, p \leftarrow \neg a, p \leftarrow \neg p\}$ . It is easy to verify that this program has two *pstable models*, which are  $\{a, p\}$  and  $\{b, p\}$ .

Since it is standard in logic programming to consider constraints, we also can define a *pstable semantics* for basic programs extended with constraints.

**Definition 2.** A basic program with constraints is a pair  $(P, R)$  such that  $P$  is a basic program and  $R$  is a set of constraints, namely that each formula in  $R$  is a constraint formula.

By a slight abuse of notation we normally write a program  $(P, R)$  just as the single program  $P \cup R$ .

**Definition 3.** Let  $(P, R)$  be a basic program with constraints. We say that  $M$  is a model of  $(P, R)$  w.r.t. the *pstable semantics* if  $M$  is a *pstable model* of  $P$  and  $M$  models  $R$ .

*Example 2.* Take the following disjunctive program with constraints:  $P = \{b \leftarrow \neg a, a \leftarrow \neg b, \leftarrow a\}$  Note that the first two rules are disjunctive rules (actually normal rules) and only the last rule is a constraint. The *pstable semantics* of the first two rules is  $\{\{a\}, \{b\}\}$ . The *pstable semantics* of the complete program is  $\{\{b\}\}$ .

We finish this subsection describing how the true negation is added. As we shall see, true negation is considered as a syntactic abbreviation and not really as a real connective. We shall use  $G'_3$  logic that do not include true negation. It is well known that this is a possible approach in some logics as, for instance in Nelson logics [13,14]. Given a program  $P$  that may include true negation and constraints, we write  $\mathcal{L}_P^e$  to denote its extended signature that consists in  $\mathcal{L}_P \cup \{-x : x \in \mathcal{L}_P\}$ . Let  $\Delta_P$  be the set of constraints  $\{(x \wedge \neg x) \rightarrow \perp : x \in \mathcal{L}_P\}$ .

<sup>1</sup> It is worth mentioning that in [11] a *pstable model* is called a  $G'_3$ -stable model.

**Definition 4.** Let  $M$  be a consistent set of literals and  $P$  a program that may include constraints (as well as true negation). We say that  $M$  is a p-answer set of  $P$  if  $M$  is a pstable model of  $P \cup \Delta_P$ , where  $P \cup \Delta_P$  is considered a basic program with signature  $\mathcal{L}_P^e$ .

Here it is important to observe that literals are considered atoms in the context of the  $G'_3$  logic. For instance, if we consider the program  $P = \{a \leftarrow \neg - a\}$  then,  $\mathcal{L}_P = \{a\}$  and  $\mathcal{L}_P^e = \{a, -a\}$ . Moreover, we can verify that  $\{a\}$  is a pstable model of  $P$  since  $\neg - a \rightarrow a$ ,  $\neg - a \models a$  and  $\{a\}$  models  $P$ .

We remark that this semantics is called *pstable* to remind that this semantics is based on a *paraconsistent logic*. For further reading on pstable semantics, refer to [10,11]. It is also worth mentioning that Sakama et al. introduce in [15] a notion of pstable models. Although Sakama's definition is very interesting and it may have some relationship with the definition given in this paper, our definition of pstable semantics is different to the definition given in [15].

### 3 Minimal Extended Generalized Pstable Models

One of the contributions of this paper corresponds to the definition of a semantics for updates consisting of a sequence of programs. The definition of such semantics is based on the notion of minimal extended generalized pstable models. So, in this section we introduce the definition of minimal extended generalized pstable models which is based on ideas from [7,2,12]. In particular in [12] it is also given a semantics for updates consisting of a sequence of programs. However, the semantics given in [12] is based on answer sets semantics [6] and the semantics given in this paper is based on pstable semantics.

We start introducing the definition of an Extended Abductive Logic (EAL) program. We can see that this definition is similar to the Definition of Abductive Logic programs in [2], but it has added a surjective function that will be used to define an order among the extended generalized pstable models of an EAL program. We also present the definition of extended generalized pstable models of an EAL program.

**Definition 5.** An extended abductive logic (EAL) program is a triple  $\langle P, A, f \rangle$  such that  $P$  is an arbitrary program,  $A$  is a set of atoms, and  $f$  is some surjective function with domain  $A$  and codomain  $\{1, \dots, N\}$ ,  $N > 0$ .

*Example 3.* We can define an EAL program  $\langle P, A, f \rangle$  where  $P$  is the following program:

$$\begin{aligned} a &\leftarrow \neg x_1^1. \\ b &\leftarrow a, \neg x_2^1. \\ -a &\leftarrow \neg x_1^2. \\ -b &\leftarrow \neg x_2^2. \\ c &\leftarrow . \end{aligned}$$

$A$  is the following set of atoms:  $\{x_1^1, x_2^1, x_1^2\}$ ; and  $f : A \rightarrow \{1, 2\}$  is the function such that  $f(x_j^i) = i$ .

**Definition 6.** Let  $\langle P, A, f \rangle$  be an EAL program,  $M$  be a set of literals, and  $\Delta \subseteq A$ . An extended generalized (EG) pstable model of  $\langle P, A, f \rangle$  is a pair  $\langle M, \Delta \rangle$  if  $M$  is a pstable model of  $P \cup \Delta$ .

*Example 4.* Let us consider the EAL program  $\langle P, A, f \rangle$  of Example 3. Table 2 shows the different EG pstable models of  $\langle P, A, f \rangle$  with their respective  $\Delta \subseteq A$ .

**Table 2.**

$\Delta$	Alternative representation of $\Delta$	$\langle M, \Delta \rangle$	IU pstable model
$\emptyset$	$\langle \emptyset, \emptyset \rangle$	no exists	no exists
$\{x_1^1\}$	$\langle \emptyset, \{x_1^1\} \rangle$	no exists	no exists
$\{x_2^1\}$	$\langle \emptyset, \{x_2^1\} \rangle$	no exists	no exists
$\{x_1^1, x_2^1\}$	$\langle \emptyset, \{x_1^1, x_2^1\} \rangle$	$\langle \{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\} \rangle$	$\{-a, -b, c\}$
$\{x_1^2, x_1^1, x_2^1\}$	$\langle \{x_1^2\}, \{x_1^1, x_2^1\} \rangle$	$\langle \{x_1^2, x_1^1, x_2^1, -a, c\}, \{x_1^2, x_1^1, x_2^1\} \rangle$	$\{-a, c\}$
$\{x_2^2, x_1^1, x_2^1\}$	$\langle \{x_2^2\}, \{x_1^1, x_2^1\} \rangle$	$\langle \{x_2^2, x_1^1, x_2^1, -b, c\}, \{x_2^2, x_1^1, x_2^1\} \rangle$	$\{-b, c\}$
$\{x_1^2, x_2^2, x_1^1, x_2^1\}$	$\langle \{x_1^2, x_2^2\}, \{x_1^1, x_2^1\} \rangle$	$\langle \{x_1^2, x_2^2, x_1^1, x_2^1, c\}, \{x_1^2, x_2^2, x_1^1, x_2^1\} \rangle$	$\{c\}$
$\{x_1^2, x_2^2, x_1^1\}$	$\langle \{x_1^2, x_2^2\}, \{x_1^1\} \rangle$	$\langle \{x_1^2, x_2^2, x_1^1, c\}, \{x_1^2, x_2^2, x_1^1\} \rangle$	$\{c\}$
$\{x_1^2, x_2^2, x_2^1\}$	$\langle \{x_1^2, x_2^2\}, \{x_2^1\} \rangle$	$\langle \{x_1^2, x_2^2, x_2^1, a, c\}, \{x_1^2, x_2^2, x_2^1\} \rangle$	$\{a, c\}$
$\{x_1^2, x_2^2\}$	$\langle \{x_1^2, x_2^2\}, \emptyset \rangle$	$\langle \{x_1^2, x_2^2, a, b, c\}, \{x_1^2, x_2^2\} \rangle$	$\{a, b, c\}$
$\{x_1^1\}$	$\langle \{x_1^1\}, \emptyset \rangle$	no exists	no exists
$\{x_2^1\}$	$\langle \{x_2^1\}, \emptyset \rangle$	no exists	no exists

Now we present an order among the EG pstable models of an EAL program.

**Definition 7.** Let  $\langle P, A, f \rangle$  be an EAL program where the codomain of  $f$  is the set  $\{1, \dots, N\}$ ,  $N > 0$ . Let  $A_i = \{a \in A \mid f(a) = i\}$ . We establish an inclusion order among the EG pstable models of  $\langle P, A, f \rangle$  as follows: Let  $\langle M_1, \Delta_1 \rangle$  and  $\langle M_2, \Delta_2 \rangle$  be EG pstable models of  $\langle P, A, f \rangle$ . We define  $\langle M_1, \Delta_1 \rangle \leq_{inclu} \langle M_2, \Delta_2 \rangle$  iff there is  $k$ ,  $1 \leq k \leq N$  such that  $(\Delta_1 \cap A_k) \subset (\Delta_2 \cap A_k)$ , and for all  $j$ ,  $k < j \leq N$ ,  $(\Delta_1 \cap A_j) = (\Delta_2 \cap A_j)$ .

*Example 5.* Let us consider the EAL program  $\langle P, A, f \rangle$  of Example 3 and its EG pstable models in Table 2. We recall that  $N = 2$  and  $f : A \rightarrow \{1, 2\}$  is the function where  $f(x_i^j) = i$ . We can verify that  $A_1 = \{x_1^1, x_2^1\}$  and  $A_2 = \{x_1^2, x_2^2\}$ . So, according to Table 2 and Definition 7, we can verify that  $\langle \{x_1^1, x_2^1, x_2^2, a, c\}, \{x_1^1, x_2^1, x_2^2\} \rangle \leq_{inclu} \langle \{x_1^1, x_2^1, x_2^2, c\}, \{x_1^1, x_2^1, x_2^2\} \rangle$ ; since there is  $k = 1$  such that  $(\{x_2^1, x_2^2\} \cap A_1) \subset (\{x_1^1, x_2^1, x_2^2\} \cap A_1)$ , i.e.,  $\{x_2^1\} \subset \{x_1^1, x_2^1\}$ , and for all  $j$ ,  $1 < j \leq 2$  we have that  $(\{x_2^1, x_2^2\} \cap A_2) = (\{x_1^1, x_2^1, x_2^2\} \cap A_2)$ , i.e.,  $\{x_1^2, x_2^2\} = \{x_2^2, x_2^2\}$ . In a similar way we can verify that  $\langle \{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\} \rangle \leq_{inclu} \langle \{x_1^1, x_2^1, x_2^2, -a, c\}, \{x_1^1, x_2^1, x_2^2\} \rangle$ , since there is  $k = 2$  such that  $(\{x_1^1, x_2^1\} \cap A_2) \subset (\{x_1^1, x_2^1, x_2^2\} \cap A_2)$ , i.e.,  $\emptyset \subset \{x_1^2\}$ , and there is no  $j$ ,  $2 < j \leq 2$ .

Alternatively and in order to make easier to understand how the order among the EG pstable models of an EAL program works, we could rewrite each subset

$\Delta$  in Table 2 as it is shown in the second column of the same table. Each subset  $\Delta$  of Table 2 is rewritten as a pair where its entries correspond to the subsets of  $\Delta$  in descendent lexicographic order with respect to the superindex of each atom in  $\Delta$ . For instance,  $\{x_2^2, x_1^1, x_2^1\}$  is rewritten as  $\{\{x_2^2\}, \{x_1^1, x_2^1\}\}$ . So, using the second column of Table 2 it is easier to verify the order of the EG pstable models. Given two EG pstable models we only verify if one of them is subset of the other with respect to a descendent lexicographic order with respect to the superindex of each atom in  $\Delta$ . For instance, considering Example 5 we can verify that  $\{\{x_2^1, x_1^2, x_2^2, a, c\}, \{x_2^1, x_1^2, x_2^2\}\} \leq_{inclu} \{\{x_1^1, x_2^1, x_2^2, c\}, \{x_1^1, x_2^1, x_2^2\}\}$  since we can see that  $\{\{x_2^1, x_1^2, x_2^2, a, c\}, \{x_2^1, x_1^2, x_2^2\}\}$  corresponds to  $\{\{x_2^1, x_2^2\}, \{x_1^1\}\}$  and  $\{\{x_1^1, x_2^1, x_2^2, c\}, \{x_1^1, x_2^1, x_2^2\}\}$  corresponds to  $\{\{x_2^1, x_2^2\}, \{x_1^1, x_2^1\}\}$ . So both pairs have the same first entry  $\{x_2^1, x_2^2\}$  but, their second entry is different and we see that  $\{x_1^1\} \subset \{x_1^1, x_2^1\}$ . In a similar way we can verify that  $\{\{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\}\} \leq_{inclu} \{\{x_1^2, x_1^1, x_2^1, -a, c\}, \{x_1^2, x_1^1, x_2^1\}\}$  since  $\emptyset \subset \{x_1^2\}$ .

Let us notice that we can also define a cardinality order among the EG pstable models of an AL program  $\langle P, A, f \rangle$ , denoted by  $\leq_{card}$ . This definition can be obtained from Definition 7 by replacing the set inclusion criterion by the set cardinality criterion. In the rest of this paper we will use only inclusion order among the EG pstable models and we will write  $\leq$  to denote this order.

It is also worth mentioning that in Definition 7 the program  $P$  is used to get the EG pstable models, although it is not used to define the order among the EG pstable models. The order among the EG pstable models is defined in terms of the subsets of  $A$  ( $\Delta_i$  and  $A_j$ ). Moreover, this order can be used to get the minimal extended generalized (MEG) pstable models of an EAL program  $\langle P, A, f \rangle$ . We shall see that a MEG pstable model is a pair  $\langle M, \Delta \rangle$ , but for all practical purposes, we are only interested in  $M$ .

**Definition 8.** Let  $\langle P, A, f \rangle$  be an EAL program.  $\langle M, \Delta \rangle$  is a minimal extended generalized (MEG) pstable model of  $\langle P, A, f \rangle$  if  $\langle M, \Delta \rangle$  is an EG pstable model of  $P$  and there is no EG pstable model  $\langle M', \Delta' \rangle$  of  $\langle P, A, f \rangle$  such that  $\langle M', \Delta' \rangle \leq \langle M, \Delta \rangle$ .

*Example 6.* Let us consider the EAL program  $\langle P, A, f \rangle$  of Example 3. According to Table 2 and Definition 8, we can verify that  $\{\{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\}\}$  is the only MEG pstable model of  $\langle P, A, f \rangle$  since there is no EG pstable model  $\langle M', \Delta' \rangle$  of  $\langle P, A, f \rangle$  such that  $\langle M', \Delta' \rangle \leq \{\{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\}\}$ .

## 4 Updates Using Minimal Extended Generalized Pstable Models

In this section we present an application of minimal extended generalized pstable models. We show how the semantics for updates consisting of a sequence of programs is given by the extended generalized pstable models.

Formally, by an update consisting of a sequence of programs, we understand a sequence  $(P_1, \dots, P_n)$  of logic programs where  $N_{P_i}$  is the number of rules in each

logic program. We say that  $\mathbf{P}$  is an update consisting of a sequence of programs over  $\mathcal{L}_{\mathbf{P}}$  if  $\mathcal{L}_{\mathbf{P}}$  represents the set of atoms occurring in  $\bigcup_{1 \leq i \leq n} P_i$ .

**Definition 9.** *Given an update consisting of a sequence of programs  $\mathbf{P} = (P_1, \dots, P_n)$  over  $\mathcal{L}_{\mathbf{P}}$ , we define the update program  $\mathbf{P}_{\circ} = P_1 \circ \dots \circ P_n$  over  $\mathcal{L}_{\mathbf{P}}^*$  (extending  $\mathcal{L}_{\mathbf{P}}$  by new abducible atoms) consisting of the following items:*

1. all constraints in  $P_1, \dots, P_{n-1}$ ,
2. for each  $r_j \in P_i, 1 \leq i \leq n-1, 1 \leq j \leq N_{P_i}$  we add the rule  $r_j \leftarrow \neg b_j^i$ , where  $b_j^i$  is an abducible (a new atom),
3. all rules  $r \in P_n$ .

An EAL program of  $\mathbf{P}$  is a triple  $\langle \mathbf{P}_{\circ}, B, f \rangle$  such that  $B$  is the set of abducibles of  $\mathbf{P}_{\circ}$ , i.e.,  $B = \{b_j^i \mid b_j^i \in \mathbf{P}_{\circ}, 1 \leq i \leq n-1, 1 \leq j \leq N_{P_i}\}$ ; and  $f : B \rightarrow \{1, \dots, n-1\}$  is the surjective function where  $f(b_j^i) = i$ .

Note that strictly following Definition 9, rule  $b \leftarrow a$  is translated into  $(b \leftarrow a) \leftarrow \neg x_2^1$ . However, this last rule is equivalent in  $G'_3$  logic to  $b \leftarrow a, \neg x_2^1$ . Additionally, the careful reader can notice that the semantics for the update program is not modified.

*Example 7.* Let  $\mathbf{P} = (P_1, P_2, P_3)$  be an update consisting of a sequence of programs over  $\mathcal{L}_{\mathbf{P}} = \{a, b, c\}$  where,

$$\begin{array}{lll}
 P_1 : & a \leftarrow . & P_2 : & -a \leftarrow . & P_3 : & c \leftarrow . \\
 & b \leftarrow a. & & -b \leftarrow . & & 
 \end{array}$$

So, the update program  $\mathbf{P}_{\circ} = P_1 \circ P_2 \circ P_3$  over  $\mathcal{L}_{\mathbf{P}}^*$  (extending  $\mathcal{L}_{\mathbf{P}}$  by new abducible atoms  $\{x_1^1, x_2^1, x_1^2, x_2^2\}$ ) is the following program:

$$\begin{array}{l}
 a \leftarrow \neg x_1^1. \\
 b \leftarrow a, \neg x_2^1. \\
 -a \leftarrow \neg x_1^2. \\
 -b \leftarrow \neg x_2^2. \\
 c \leftarrow .
 \end{array}$$

The EAL program of  $\mathbf{P}$  is the triple  $\langle \mathbf{P}_{\circ}, B, f \rangle$ , where  $B = \{x_1^1, x_2^1, x_1^2, x_2^2\}$ ; and  $f : B \rightarrow \{1, 2\}$  is the function  $f(x_j^i) = i$ .

Now we shall see how the update pstable models of an update sequence with respect to  $\circ$ , denoted by  $\circ$ -update pstable models, can be gotten from the MEG pstable models of a given EAL program.

**Definition 10.** *Let  $\mathbf{P} = (P_1, \dots, P_{n-1}, P_n)$  be an update consisting of a sequence of programs over the set of atoms  $\mathcal{L}_{\mathbf{P}}$ . Let  $\langle M', \Delta' \rangle$  be a MEG pstable model of the EAL program  $\langle \mathbf{P}_{\circ}, B, f \rangle$ . Then,  $M$  is an update  $\circ$ -pstable model of  $\mathbf{P}$  if and only if  $M = M' \cap \mathcal{L}_{\mathbf{P}}$ .*

*Example 8.* Let us consider the update consisting of a sequence of programs  $\mathbf{P} = (P_1, P_2, P_3)$  of Example 7. In Example 6 we verified that  $\langle \{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\} \rangle$  is the only MEG stable model of  $\langle \mathbf{P}_{\circ}, B, f \rangle$ . So, according to Definition 10, we can verify that  $\{-b, -a, c\}$  is the only  $\circ$ -update pstable model of  $\mathbf{P}$ .

Now, let us consider Example 1 from [9] as another example to illustrate Definition 10.

*Example 9.* Let us consider the update consisting of a sequence of programs  $\mathbf{P} = (P_1, P_2)$  where

$$\begin{array}{ll}
 P_1 : \text{sleep} \leftarrow \text{night}, \neg \text{watchTv}, \neg \text{other}. & P_2 : \neg \text{tvOn} \leftarrow \text{powerFailure}. \\
 \text{night} \leftarrow . & \neg \text{tvOn} \leftarrow \text{assignmentDue}, \text{working}. \\
 \text{tvOn} \leftarrow \neg \text{tvBroke}. & \text{assignmentDue} \leftarrow . \\
 \text{watchTv} \leftarrow \text{tvOn}. & \text{working} \leftarrow . \\
 \text{other} \leftarrow \text{working}. &
 \end{array}$$

So, the update program  $\mathbf{P}_\circ = P_1 \circ P_2$  over  $\mathcal{L}_{\mathbf{P}}^*$  (extending  $\mathcal{L}_{\mathbf{P}}$  by new abducible atoms  $\{x_1^1, x_2^1, x_3^1, x_4^1\}$ ) is the following program:

$$\begin{array}{l}
 \text{sleep} \leftarrow \text{night}, \neg \text{watchTv}, \neg \text{other}, \neg x_1^1. \\
 \text{night} \leftarrow, \neg x_2^1. \\
 \text{tvOn} \leftarrow \neg \text{tvBroke}, \neg x_3^1. \\
 \text{watchTv} \leftarrow \text{tvOn}, \neg x_4^1. \\
 \neg \text{tvOn} \leftarrow \text{powerFailure}. \\
 \neg \text{tvOn} \leftarrow \text{assignmentDue}, \text{working}. \\
 \text{assignmentDue} \leftarrow . \\
 \text{working} \leftarrow . \\
 \text{other} \leftarrow \text{working}.
 \end{array}$$

The EAL program of  $\mathbf{P}$  is the triple  $\langle \mathbf{P}_\circ, B, f \rangle$ , where  $B = \{x_1^1, x_2^1, x_3^1, x_4^1\}$ ; and  $f : B \rightarrow \{1\}$  is the function,  $f(x_j^1) = i$ . We can verify that the only MEG pstable model of  $\langle \mathbf{P}_\circ, B, f \rangle$  is  $\langle \{x_3^1, \text{night}, \text{other}, \text{assignmentDue}, \text{working}, \neg \text{tvOn}\}, \{x_3^1\} \rangle$ . Finally, according to Definition 10, the only  $\circ$ -update pstable model of  $\mathbf{P}$  which coincides with the result of Example 1 in [9] is:

$$\langle \text{night}, \text{other}, \text{assignmentDue}, \text{working}, \neg \text{tvOn} \rangle.$$

## 5 Alternative Semantics for Update Sequences Based on Minimal Pstable Models

In this section we introduce an alternative to the semantics for update sequences given in section 4. This semantics can be considered as another application of minimal extended generalized pstable models. We can make the alternative semantics clear with the following example.

*Example 10.* Let  $\mathbf{P} = (P_1, P_2)$  be an update sequence over  $\mathcal{L}_{\mathbf{P}} = \{a, b\}$  where,  $P_1$  and  $P_2$  are the following logic programs,

$$\begin{array}{ll}
 P_1 : & b \leftarrow . \\
 & a \leftarrow b. \\
 P_2 : & \neg a \leftarrow .
 \end{array}$$

Using the semantics given in section 4 the update pstable models are  $\{-a, b\}$  and  $\{-a\}$ . However, using the alternative semantics we expect to get only  $\{-a, b\}$  since it has more information than  $\{-a\}$ .

So, we present an alternative semantics based on MEG pstable models that chooses the pstable models with maximal cardinality. It is worth mentioning that by applying the semantics given in [5] to the update sequence in Example 10 we also get  $\{-a, b\}$ .

The update operator of this alternative semantics is represented as  $\mathcal{O}'$ .

**Definition 11.** *Let  $\mathbf{P} = (P_1, \dots, P_n)$  be an update sequence over the set of atoms  $\mathcal{L}_{\mathbf{P}}$ . Then,  $M$  is an  $\mathcal{O}'$ -update pstable model of  $\mathbf{P}$ , if  $M$  is an  $\mathcal{O}$ -update pstable model of  $\mathbf{P}$  and it is maximal among the  $\mathcal{O}$ -update pstable models of  $\mathbf{P}$  with respect to inclusion order.*

## 6 Conclusions and Future Work

In this paper we presented two update semantics for sequences of programs which are based on pstable semantics and represent an alternative to the update semantics given in [7,2,12,5,19] which are based on answer sets semantics [6]. Since the semantics given in this paper are based on pstable semantics, it allows us to keep a compromise with classical logic which is a logic that currently has many applications.

As future work we plan to analyze which of those properties described in [5,9] hold in our update approach. We also plan to give a general definition of those properties.

## References

1. Alferes, J.J., Pereira, L.M.: Logic programming updating - a guided approach. In: Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II, pp. 382–412. Springer, London (2002)
2. Balduccini, M., Gelfond, M.: Logic Programs with Consistency-Restoring Rules. In: Doherty, P., McCarthy, J., Williams, M.-A. (eds.) International Symposium on Logical Formalization of Commonsense Reasoning. AAAI 2003 Spring Symposium Series (March 2003)
3. Carballido, J., Osorio, M., Arrazola, J.: Equivalence in the  $G'3$ -stable semantics. Accepted to appear in Research of Computing Science Journal (November 2007)
4. Delgrande, J., Schaub, T., Tompits, H.: A preference-based framework for updating logic programs. In: Baral, C., Brewka, G., Schlipf, J. (eds.) LPNMR 2007. Proceedings of the Ninth International Conference on Logic Programming and Nonmonotonic Reasoning. LNCS (LNAI), vol. 4483, pp. 71–83. Springer, Heidelberg (2007)
5. Eiter, T., Fink, M., Sabbatini, G., Tompits, H.: On properties of update sequences based on causal rejection. Theory and Practice of Logic Programming 2(6), 711–767 (2002)
6. Gelfond, M., Lifschitz, V.: The Stable Model Semantics for Logic Programming. In: Kowalski, R., Bowen, K. (eds.) 5th Conference on Logic Programming, pp. 1070–1080. MIT Press, Cambridge (1988)
7. Kakas, A.C., Mancarella, P.: Generalized stable models: a semantics for abduction. In: Proceedings of ECAI-90, pp. 385–391. IOS Press, Amsterdam (1990)



8. Osorio, M., Arrazola, J., Carballido, J., Estrada, O.: Aan axiomatization og  $\mathcal{G}3$ . In: LoLaCOM 2006. Proceedings of the Workshop in Logic, Language and Computation 2006, Apizaco, Tlaxcala, México, November 13-14 2006 (2006), [http://ftp.informatik.rwth--aachen.de/Publications/CEUR--WS/Vol--220/LoLaCOM\\_06\\_05.pdf](http://ftp.informatik.rwth--aachen.de/Publications/CEUR--WS/Vol--220/LoLaCOM_06_05.pdf)
9. Osorio, M., Cuevas, V.: Updates in answer set programming: An approach based on basic structural properties. *Theory and Practice of Logic Programming* 7(04), 451–479 (2007)
10. Osorio, M., Navarro, J.A., Arrazola, J., Borja, V.: Ground nonmonotonic modal logic S5: New results. *Journal of Logic and Computation* 15(5), 787–813 (2005)
11. Osorio, M., Navarro, J.A., Arrazola, J., Borja, V.: Logics with common weak completions. *Journal of Logic and Computation* 16(6), 867–890 (2006)
12. Osorio, M., Zepeda, C.: A semantics for updates consisting of a sequence of program In: CONIELECOMP 2007. Electronic Proceedings of the 17th International Conference on Electronics, Communications, and Computers, Puebla, Mexico (2007) (ISBN 0-7695-2799-X)
13. Rasiowa, H.: N -lattices and constructive logic with strong negation. *Fundamenta Mathematicae* 46, 61–80 (1958)
14. Rasiowa, H.: *An Algebraic Approach to Non-Classical Logics*. American Elsevier Publishing Company, New York (1974)
15. Sakama, C., Inoue, K.: Paraconsistent stable semantics for extended disjunctive programs. *Journal of Logic and Computation* 5(3), 265–285 (1995)
16. Zacarias, F., Galindo, M.O., Guadarrama, J.C.A., Dix, J.: Updates in Answer Set Programming based on structural properties. In: Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning. Dresden University Technical Report, Corfu, Greece, TU-Dresden, Fakultt Informatik, pp. 213–219 (2005)
17. Zepeda, C., Osorio, M., Nieves, J.C., Solmon, C., Sol, D.: Applications of preferences using answer set programming. In: ASP 2005. Answer Set Programming: Advances in Theory and Implementation, University of Bath, UK, pp. 318–332 (July 2005)

# PStable Semantics for Possibilistic Logic Programs

Mauricio Osorio<sup>1</sup> and Juan Carlos Nieves<sup>2</sup>

<sup>1</sup> Universidad de las Américas - Puebla  
CENTIA, Sta. Catarina Mártir, Cholula, Puebla, 72820 México  
osorionmauri@googlemail.com

<sup>2</sup> Universitat Politècnica de Catalunya  
Software Department (LSI)  
c/Jordi Girona 1-3, E08034, Barcelona, Spain  
jcnieves@lsi.upc.edu

**Abstract.** Uncertain information is present in many real applications *e.g.*, medical domain, weather forecast, *etc.* The most common approaches for leading with this information are based on probability however some times; it is difficult to find suitable probabilities about some events. In this paper, we present a possibilistic logic programming approach which is based on possibilistic logic and PStable semantics. Possibilistic logic is a logic of uncertainty tailored for reasoning under incomplete evidence and Pstable Semantics is a solid semantics which emerges from the fusion of non-monotonic reasoning and logic programming; moreover it is able to express answer set semantics, and has strong connections with paraconsistent logics.

## 1 Introduction

To find a representation of the information under uncertainty has been subject of much debate. For those steeped in probability, there is only one appropriate model for numeric uncertainty, and that is probability. But probability has its problems. For one thing, the numbers are not always available. For another, the commitment to numbers means that any two events must be comparable in terms of probability: either one event is more probable than the other, or they have equal probability [4]. In fact, in [7], McCarthy and Hayes pointed out that attaching probabilities to a statement has some objections. For instance:

The information necessary to assign numerical probabilities is not ordinary available. Therefore, a formalism that required numerical probabilities would be epistemologically inadequate [7].

Hence it is not surprising that many other representations of uncertainty have been considered in the literature. For instance in the MYCIN project which is one of the clearest representatives of the experimental side of Artificial Intelligence (IA) was shown that probability theory has limitations for developing automated assistance for medical diagnosis [2]. In this project, it was adopted a less formal model. This model uses estimates provided by expert physicians that reflect the tendency of a piece of evidence to prove or disprove a hypothesis. The syntax adopted by MYCIN was based on IF-THEN rules with certainty factors. The following is an English version of one of MYCIN's rules:

**IF** the infection is primary-bacteremia (**a**)  
**AND** the site of the culture is one of the sterile sites (**b**)  
**AND** the suspected portal of entry is the gastrointestinal tract (**c**)  
**THEN** there is suggestive evidence (0.7) that infection is bacteroid (**d**).

The 0.7 is roughly the certainty that the conclusion will be true given the evidence. If the evidence is uncertain the certainties of the bits of evidence will be combined with the certainty of the rule to give the certainty of the conclusion.

John McCarthy pointed out in his seminal paper [6] that the MYCIN's major innovation over many previous expert systems was that it uses measures of uncertainty (not probabilities) for its diagnoses and the fact that it is prepared to explain its reasoning to the physician. We can say that MYCIN was one of the most successful projects at its time; however, it seems that AI's community has taken few lessons from MYCIN's experience for developing new intelligence support systems.

One of the main problems of MYCIN was that it developed a monotonic reasoning about its diagnoses. Then to update the medical knowledge in order to improve its diagnoses was a problem. Nowadays, logic programming and non-monotonic reasoning are solid areas in AI. For instance, during the last two decades, one of the most successful logic programming approaches has been Answer Set Programming (ASP). ASP is the realization of much theoretical work on Non-monotonic Reasoning and Artificial Intelligence applications. It represents a new paradigm for logic programming that allows, using the concept of *negation as failure*, to handle problems with default knowledge and produce non-monotonic reasoning [1].

In [9], it was proposed a possibilistic logic programming framework for reasoning under uncertainty. It is a combination between Answer Set Programming (ASP) [1] and Possibilistic Logic [3]. This framework is able to deal with reasoning that is at the same time non-monotonic and uncertain. Since this framework was defined for normal programs, it was generalized in [10] for capturing possibilistic disjunctive programs and allowing the encoding of uncertain information by using either numerical values or relative likelihoods.

We can accept that the language's expressiveness of the Possibilistic Answer Set Programming approach (PASP) presented in [9,10] is rich enough for capturing a wide family of problems where one have to confront with incomplete information and uncertain information. For instance, the MYCIN's rule presented previously can be expressed as follows:

$$07 : d \leftarrow a, b, c.$$

where  $a, b, c, d$  are propositional atoms whose intended meanings are described in the previous MYCIN's rule.

PASP could be considered as a good option for developing new intelligence support systems like MYCIN system such that the support systems could perform non-monotonic reasoning. However it is obvious that an intelligence support system always must have an answer to any query to its knowledge base. Since PASP was defined as an ASP's extension, there are some possibilistic logic programs which have no possibilistic answer sets. For instance, a single possibilistic clause as  $\alpha : a \leftarrow \text{not } a$  does not have a possibilistic answer set. In fact, the existence of one clause of this form will affect all the possibilistic knowledge base such that all the possibilistic knowledge base

will not have a possibilistic answer set. It is quite obvious that this situation could not be permitted in an intelligence support system like MYCIN.

In this paper, we define a possibilistic logic programming semantics called *possibilistic pstable semantics*. This semantics is based on pstable semantics [11,12]. Pstable semantics emerges from the fusion of paraconsistent logics and ASP. This semantics is able to capture ASP; moreover it is less sensitive than the answer set semantics.

Like in possibilistic answer set semantics, possibilistic pstable semantics is based on possibilistic logic. Possibilistic logic is a type of logic of uncertainty tailored for reasoning under incomplete evidence and partially inconsistent knowledge. At the syntactic level it handles formulae of propositional or first-order logic to which are attached degrees of necessity. The degree of necessity (or certainty) of a formula expresses to what extent the available evidence entails the truth of this formula [3]. We argue that possibilistic logic is an excellent approximation of the approach adopted by MYCIN system which showed that it is practical in real applications.

It is worth mentioning that possibilistic pstable semantics is close related to possibilistic logic. For instance, it has the property that given a possibilistic logic program  $P$ , if  $P \vdash_{PL} (x \alpha)$  then  $P$  is equivalent to  $P \cup \{(x \alpha)\}$  under the possibilistic pstable semantics.

The rest of the paper is divided as follows: In §2 some basic definitions of possibilistic logic and pstable semantics are presented. In §3 the syntax of our possibilistic framework is presented. In §4 the possibilistic pstable semantics is defined. Finally in the last section, we present our conclusions.

## 2 Background

In this section, we define some basic concepts of Possibilistic Logic and Pstable models. We assume familiarity with basic concepts in classic logic and in semantics of logic programs *e.g.*, interpretation, model, *etc.* A good introductory treatment of these concepts can be found in [18].

### 2.1 Possibilistic Logic

A necessity-valued formula is a pair  $(\varphi \alpha)$  where  $\varphi$  is a classical logic formula and  $\alpha \in (0, 1]$  is a positive number. The pair  $(\varphi \alpha)$  expresses that the formula  $\varphi$  is certain at least to the level  $\alpha$ , *i.e.*  $N(\varphi) \geq \alpha$ , where  $N$  is a necessity measure modeling our possibly incomplete state knowledge [3].  $\alpha$  is not a probability (like it is in probability theory) but it induces a certainty (or confidence) scale. This value is determined by the expert providing the knowledge base. A necessity-valued knowledge base is then defined as a finite set (*i.e.* a conjunction) of necessity-valued formulae.

Dubois *et al.* [3] introduced a formal system for necessity-valued logic which is based on the following axioms schemata (propositional case):

$$(A1) (\varphi \rightarrow (\psi \rightarrow \varphi) 1)$$

$$(A2) ((\varphi \rightarrow (\psi \rightarrow \xi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \xi)) 1)$$

$$(A3) ((\neg\varphi \rightarrow \neg\psi) \rightarrow ((\neg\varphi \rightarrow \psi) \rightarrow \varphi) 1)$$

As in classic logic, the symbols  $\neg$  and  $\rightarrow$  are considered primitive connectives, then connectives as  $\vee$  and  $\wedge$  are defined as abbreviations of  $\neg$  and  $\rightarrow$ . Now the inference rules for the axioms are:

- (GMP)  $(\varphi \alpha), (\varphi \rightarrow \psi \beta) \vdash (\psi \min\{\alpha, \beta\})$
- (S)  $(\varphi \alpha) \vdash (\varphi \beta)$  if  $\beta \leq \alpha$

According to Dubois *et al.*, basically we need a complete lattice in order to express the levels of uncertainty in Possibilistic Logic. Dubois *et al.*, extended the axioms schemata and the inference rules for considering partially ordered sets. We shall denote by  $\vdash_{PL}$  the inference under Possibilistic Logic without paying attention if the necessity-valued formulae are using either a totally ordered set or a partially ordered set for expressing the levels of uncertainty.

The problem of inferring automatically the necessity-value of a classical formula from a possibilistic base was solved by an extended version of *resolution* for possibilistic logic (see [3] for details).

## 2.2 Syntax: Logic Programs

The language of a propositional logic has an alphabet consisting of

- (i) proposition symbols:  $p_0, p_1, \dots$
- (ii) connectives :  $\vee, \wedge, \leftarrow, \neg, not, \perp$
- (iii) auxiliary symbols :  $(, )$ .

where  $\vee, \wedge, \leftarrow$  are 2-place connectives,  $\neg, not$  are 1-place connective and  $\perp$  is 0-place connective. The proposition symbols,  $\perp$ , and the propositional symbols of the form  $\neg p_i$  ( $i \geq 0$ ) stand for the indecomposable propositions, which we call *atoms*, or *atomic propositions*. The negation sign  $\neg$  is regarded as the so called *strong negation* by the ASP's literature and the negation *not* as the *negation as failure*. A literal is an atom,  $a$ , or the negation of an atom *not*  $a$ . Given a set of atoms  $\{a_1, \dots, a_n\}$ , we write *not*  $\{a_1, \dots, a_n\}$  to denote the set of literals  $\{not\ a_1, \dots, not\ a_n\}$ .

An extended normal clause,  $C$ , is denoted:

$$a \leftarrow a_1, \dots, a_j, not\ a_{j+1}, \dots, not\ a_n$$

where  $n \geq 0$ ,  $a$  is an atom and each  $a_i$  is an atom. When  $n = 0$  the clause is an abbreviation of  $a$ . An extended normal program  $P$  is a finite set of extended normal clauses. By  $\mathcal{L}_P$ , we denote the set of atoms in the language of  $P$ .

We will manage the strong negation ( $\neg$ ), in our logic programs, as it is done in ASP [1]. Basically, it is replaced each atom of the form  $\neg a$  by a new atom symbol  $a'$  which does not appear in the language of the program. For instance, let  $P$  be the extended normal program:

$$a \leftarrow q. \quad \neg q \leftarrow r. \quad q. \quad r.$$

Then replacing the atom  $\neg q$  by a new atom symbol  $q'$ , we will have:

$$a \leftarrow q. \quad q' \leftarrow r. \quad q. \quad r.$$

In order not to allow inconsistent answer sets in the ASP's programs, usually it is added a normal clause of the form  $f \leftarrow q, q', f$  such that  $f \notin \mathcal{L}_P$ . We will omit this clause in order to allow an inconsistent level in our possibilistic pstable models. However the user could add this clause without losing generality.

Sometimes we denote an extended normal clause  $C$  by  $a \leftarrow \mathcal{B}^+, \text{not } \mathcal{B}^-$ , where  $\mathcal{B}^+$  contains all the positive body literals and  $\mathcal{B}^-$  contains all the negative body literals. When  $\mathcal{B}^- = \emptyset$ , the clause is called definite clause. A set of definite clauses is called a definite logic program.

### 2.3 Pstable Semantics

First to definite pstable semantics, we define some basic concepts. Logic consequence in classic logic is denoted by  $\vdash$ . Given a set of proposition symbols  $S$  and a theory (a set of well-formed formulae)  $\Gamma$ , if  $\Gamma \vdash S$  if and only if  $\forall s \in S \Gamma \vdash s$ . When we treat a logic program as a theory, each negative literal  $\text{not } a$  is replaced by  $\sim a$  such that  $\sim$  is regarded as the classical negation in classic logic. Given a normal program  $P$ , if  $M \subseteq \mathcal{L}_P$ , we write  $P \Vdash M$  when:  $P \vdash M$  and  $M$  is a classical 2-valued model of  $P$  (i.e. atoms in  $M$  are set to true, and atoms not in  $M$  to false; the set of atoms is a classical model of  $P$  if the induced interpretation evaluates  $P$  to true).

Pstable semantics is defined in terms of a single reduction which is defined as follows:

**Definition 1.** [U2] Let  $P$  be a normal program and  $M$  a set of literals. We define

$$RED(P, M) := \{l \leftarrow \mathcal{B}^+, \text{not } (\mathcal{B}^- \cap M) \mid l \leftarrow \mathcal{B}^+, \text{not } \mathcal{B}^- \in P\}$$

Let us consider the set of atoms  $M_1 := \{a, b\}$  and the following normal program  $P_1$ :

$$\begin{aligned} a &\leftarrow \text{not } b, \text{not } c. \\ a &\leftarrow b. \\ b &\leftarrow a. \end{aligned}$$

We can see that  $RED(P, M)$  is:

$$\begin{aligned} a &\leftarrow \text{not } b. \\ a &\leftarrow b. \\ b &\leftarrow a. \end{aligned}$$

By considering the reduction  $RED$ , it is defined the semantics *pstable* for normal programs.

**Definition 2.** [U2] Let  $P$  be a normal program and  $M$  a set of atoms. We say that  $M$  is a pstable model of  $P$  if  $RED(P, M) \Vdash M$ . We use *Pstable* to denote the semantics operator of pstable models.

Let us consider again  $M_1$  and  $P_1$  in order to illustrate the definition. We want to verify whether  $M_1$  is a pstable model of  $P_1$ . First, we can see that  $M_1$  is a model of  $P_1$ , i.e.  $\forall C \in P_1, M_1$  evaluates  $C$  to true. Now, we have to prove each atom of  $M_1$  from  $RED(P_1, M_1)$  by using classical inference, i.e.  $RED(P_1, M_1) \vdash M_1$ . Let us consider the proof of the atom  $a$ , which belongs to  $M_1$ , from  $RED(P_1, M_1)$ .

1.  $(a \vee b) \rightarrow ((b \rightarrow a) \rightarrow a)$  Tautology
2.  $\sim b \rightarrow a$  Premise from  $RED(P_1, M_1)$
3.  $a \vee b$  From 2 by logical equivalency
4.  $(b \rightarrow a) \rightarrow a$  From 1 and 3 by Modus Ponens
5.  $b \rightarrow a$  Premise from  $RED(P_1, M_1)$
6.  $a$  From 4 and 5 by Modus Ponens

Remember that the formula  $\sim b \rightarrow a$  corresponds to the normal clause  $a \leftarrow not\ b$  which belongs to the program  $RED(P_1, M_1)$ . The proof for the atom  $b$ , which also belongs to  $M_1$ , is similar. Then we can conclude that  $RED(P_1, M_1) \Vdash M_1$ . Hence,  $M_1$  is a *pstable model* of  $P_1$ .

### 3 Possibilistic Normal Logic Programs

In this section, we introduce our possibilistic logic programming framework. We shall start by defining the syntax of a valid program and some relevant concepts, after that we shall define the semantics for the possibilistic normal logic programs. In whole paper, we will consider finite lattices. This convention was taken based on the assumption that in real applications rarely we will have an infinite set of labels for expressing the incomplete state of a knowledge base.

#### 3.1 Syntax

First of all, we start defining some relevant concepts<sup>1</sup>. A *possibilistic atom* is a pair  $p = (a, q) \in \mathcal{A} \times Q$ , where  $\mathcal{A}$  is a finite set of atoms and  $(Q, \leq)$  is a lattice. We apply the projection  $*$  over  $p$  as follows:  $p^* = a$ . Given a set of possibilistic atoms  $S$ , we define the generalization of  $*$  over  $S$  as follows:  $S^* = \{p^* | p \in S\}$ . Given a lattice  $(Q, \leq)$  and  $S \subseteq Q$ ,  $LUB(S)$  denotes the least upper bound of  $S$  and  $GLB(S)$  denotes the greatest lower bound of  $S$ . Three basic operations between sets of possibilistic atoms are formalized as follows:

**Definition 3.** Let  $\mathcal{A}$  be a finite set of atoms and  $(Q, \leq)$  be a lattice. Consider  $\mathcal{PS} = 2^{\mathcal{A} \times Q}$  the finite set of all the possibilistic atom sets induced by  $\mathcal{A}$  and  $Q$ .  $\forall A, B \in \mathcal{PS}$ , we define.

$$\begin{aligned}
 A \sqcap B &= \{(x, GLB\{q_1, q_2\}) | (x, q_1) \in A \wedge (x, q_2) \in B\} \\
 A \sqcup B &= \{(x, q) | (x, q) \in A \text{ and } x \notin B^*\} \cup \\
 &\quad \{(x, q) | x \notin A^* \text{ and } (x, q) \in B\} \cup \\
 &\quad \{(x, LUB\{q_1, q_2\}) | (x, q_1) \in A \text{ and } (x, q_2) \in B\}. \\
 A \sqsubseteq B &\iff A^* \subseteq B^*, \text{ and } \forall x, q_1, q_2, \\
 &\quad (x, q_1) \in A \wedge (x, q_2) \in B \text{ then } q_1 \leq q_2.
 \end{aligned}$$

**Proposition 1.**  $(\mathcal{PS}, \sqsubseteq)$  is a complete lattice.

Now, we define the syntax of a valid possibilistic normal logic program. Let  $(Q, \leq)$  be a lattice. A possibilistic normal clause is of the form:

$$r := (\alpha : a \leftarrow \mathcal{B}^+, not\ \mathcal{B}^-)$$

<sup>1</sup> Some concepts presented in this subsection extend some terms presented in [9].

where  $\alpha \in Q$ . The projection  $*$  over the possibilistic clause  $r$  is:  $r^* = a \leftarrow \mathcal{B}^+$ , not  $\mathcal{B}^-$ .  $n(r) = \alpha$  is a necessity degree representing the certainty level of the information described by  $r$ .

A possibilistic normal logic program  $P$  is a tuple of the form  $\langle (Q, \leq), N \rangle$ , where  $(Q, \leq)$  is a lattice and  $N$  is a finite set of possibilistic normal clauses. The generalization of the projection  $*$  over  $P$  is as follows:  $P^* = \{r^* | r \in N\}$ . Notice that  $P^*$  is an extended normal program. When  $P^*$  is a definite program,  $P$  is called a possibilistic definite logic program.

In order to illustrate a possibilistic normal logic program, let us consider the following scenario:

*Example 1.* Let us suppose that a patient suffering from certain symptoms takes a blood test, and that the results show the presence of a bacterium of a certain category in his blood. There are two types of bacteria in this category, and the blood test *does not pinpoint* whether the bacteria present in the blood is either streptococcus viridans or X. The problem is that if the bacterium is streptococcus viridans the patient have to be treated by antibiotics of large spectrum because streptococcus viridans suggests *endocarditis*. However, the doctor tries not to prescribe antibiotics of large spectrum, because they are harmful to the immune system. Then, the doctor in this case must evaluate each potential choice, where each potential choice has different levels of uncertainty<sup>2</sup>.

In order to encode this scenario let us consider the lattice  $\langle \{Open, Supported, Plausible, Supported, Probable, Confirmed, Certain\}, \leq \rangle$ , where the following relations hold:  $Open \leq Supported$ ,  $Supported \leq Plausible$ ,  $Supported \leq Probable$ ,  $Probable \leq Confirmed$ ,  $Plausible \leq Confirmed$ , and  $Confirmed \leq Certain$ . The elements of this lattice represent relative likelihoods which will be used for encoding the uncertainty of our scenario. Then, we could model the doctor's beliefs as follows: One doctor's belief is that it is *confirmed* that the patient has a bacterium of category  $n$ . Then, this belief could be encoded by:

*confirmed* : *category\_n*.

Another doctor's belief is that the category  $n$  *implies two possible bacteria*. Then it could be encoded by:

*certain* : *streptococcus\_viridans*  $\leftarrow$  *category\_n*, not *bacterium\_x*.

*certain* : *bacterium\_x*  $\leftarrow$  *category\_n*, not *streptococcus\_viridans*.

Now, if the bacterium is *streptococcus\_viridans*, then the patient *have to be* treated by antibiotics of large spectrum.

*certain* : *antibiotics\_large\_spectrum*  $\leftarrow$  *streptococcus\_viridans*.

If the bacteria is  $x$ , then the patient *could be* treated without antibiotics of large spectrum.

*probable* : *alternative\_treatment*  $\leftarrow$  *bacterium\_x*.

One of the main doctor's belief is that he should not use antibiotics of large spectrum if it has not been established that there is not another alternative treatment.

<sup>2</sup> This example is an adaptation of Example 6 from [5].



*plausible* :  $\neg \text{antibiotics\_large\_spectrum} \leftarrow \text{not } \neg \text{alternative\_treatment}$ .  
*plausible* :  $\neg \text{alternative\_treatment} \leftarrow \text{not } \neg \text{antibiotics\_large\_spectrum}$ .

We can appreciate the use of relative likelihoods could facilitate the modeling of incomplete states of a belief.

## 4 Pstable Semantics and Possibilistic Programs

In this section, we define the possibilistic pstable semantics. In order to define pstable semantics for possibilistic normal programs, let us define the following single reduction based on Definition 4:

**Definition 4.** Let  $P$  be a possibilistic normal program and  $M$  a set of literals. We define

$$PRED(P, M) := \{(\alpha : l \leftarrow \mathcal{B}^+, \text{not } (\mathcal{B}^- \cap M)) \mid (\alpha : l \leftarrow \mathcal{B}^+, \text{not } \mathcal{B}^-) \in P\}$$

Let us consider the following example.

*Example 2.* First, let  $S$  be the set  $\{(a, 0.6), (b, 0.7)\}$  and  $P_1$  be the following possibilistic normal program where the possibilistic clauses are built under the lattice  $Q := (\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}, \leq)$ :

0.7 :  $a \leftarrow \text{not } b, \text{not } c$ .  
 0.6 :  $a \leftarrow b$ .  
 0.8 :  $b \leftarrow a$ .

Then, the program  $PRED(P_1, M)$  is:

0.7 :  $a \leftarrow \text{not } b$ .  
 0.6 :  $a \leftarrow b$ .  
 0.8 :  $b \leftarrow a$ .

**Definition 5 (Possibilistic Pstable Semantics).** Let  $P$  be a possibilistic normal logic program and  $M$  be a set of possibilistic atoms such that  $M^*$  is a pstable model of  $P^*$ . We say that  $M$  is a possibilistic pstable model of  $P$  if and only if  $PRED(P, M) \vdash_{PL} M$  and  $\nexists M'$  such that  $M' \neq M$ ,  $PRED(P, M) \vdash_{PL} M'$  and  $M \sqsubseteq M'$ .

*Example 3.* Let  $P_1$  be the possibilistic program of Example 2 and  $S := \{(a, 0.6), (b, 0.7)\}$ . We have already seen that  $PRED(P_1, S)$  is:

0.7 :  $a \leftarrow \text{not } b$ .  
 0.6 :  $a \leftarrow b$ .  
 0.8 :  $b \leftarrow a$ .

Then, we want to know if  $S$  is a possibilistic pstable models of  $P_1$ . First of all, we have already seen in Section 2.3 that  $S^*$  is a pstable models of  $P_1^*$ . Hence, we have to construct a proof in possibilistic logic for  $(a, 0.6)$  and  $(b, 0.7)$ . Let us consider the proof for the possibilistic atom  $(a, 0.6)$ :

<sup>3</sup>  $\leq$  is the traditional relation between rational numbers.

1.  $(a \vee b) \rightarrow ((b \rightarrow a) \rightarrow a)$  1 Tautology
2.  $\sim b \rightarrow a$  0.7 Premise from  $PRED(P_1, M_1)$
3.  $a \vee b$  0.7 From 2 by possibilistic logical equivalency
4.  $(b \rightarrow a) \rightarrow a$  0.7 From 1 and 3 by GMP
5.  $b \rightarrow a$  0.6 Premise from  $PRED(P_1, M_1)$
6.  $a$  0.6 From 4 and 5 by GMP

The proof for  $(b, 0.7)$  is similar to the proof of  $(a, 0.6)$ . Notice that  $\nexists S'$  such that  $PRED(P_1, S) \vdash_{PL} S'$  and  $S \sqsubseteq S'$ . Therefore, we can conclude that  $S$  is a *possibilistic pstable models* of  $P_1$ .

It is worth mentioning that the possibilistic program  $P_1$  is an example where the possibilistic pstable semantics is different to the possibilistic stable semantics [9] and the possibilistic answer set semantics [10]. In fact,  $P_1$  has no possibilistic stable model neither possibilistic answer set.

In order to complete Example 1 we can see that the scenario has two possibilistic pstable models:

1.  $\{(category\_n, confirmed), (streptococcus\_viridans, certain), (antibiotics\_large\_spectrum, certain), (\neg alternative\_treatment, plausible)\}$
2.  $\{(category\_n, confirmed), (bacterium\_x, certain), (alternative\_treatment, probable), (\neg antibiotics\_large\_spectrum, plausible)\}$

Notice that each one suggests a treatment depending of the bacterium. In this example the possibilistic pstable semantics coincides with the possibilistic answer semantics [10].

When the possibilistic normal program is built under a totally ordered set the possibilistic pstable semantics coincides with the possibilistic answer set semantics and the possibilistic stable semantics.

**Lemma 1.** *Let  $P := \langle (Q, \leq), N \rangle$  be a possibilistic normal program such that  $(Q, \leq)$  is a totally ordered set.  $S$  is a possibilistic pstable model of  $P$  iff  $S$  is a possibilistic stable model of  $P$  iff  $S$  is a possibilistic answer set of  $S$ .*

An outstanding property of the possibilistic pstable semantics is that this semantics support a kind of monotony *w.r.t.* the inference under possibilistic logic. In order to formalize this property, we will say that  $P$  is equivalent to  $P'$  under the possibilistic pstable semantics if and only if any possibilistic pstable model of  $P$  is also a possibilistic pstable model of  $P'$  and vice versa.

**Lemma 2.** *Let  $P$  be a possibilistic normal program. If  $P \vdash_{PL} (x \alpha)$  then  $P$  is equivalent to  $P \cup \{(x \alpha)\}$  under the possibilistic pstable semantics.*

Notice that the possibilistic answer set semantics [10] and the possibilistic stable semantics [9] do not satisfy that if  $P \vdash_{PL} (x \alpha)$ , then  $P$  is equivalent to  $P \cup \{(x \alpha)\}$  under either the possibilistic answer set semantics or the possibilistic stable semantics. In order to show this, let us consider the single possibilistic logic program  $P$ :

$$\alpha : a \leftarrow not a$$

It is clear that  $P \vdash_{PL} (a \ \alpha)$ .  $P$  has no possibilistic stable model neither possibilistic answer set. However  $P \cup \{(a \ \alpha)\}$  has a possibilistic stable model and a possibilistic answer set which is the same in both cases and is  $\{(a, \alpha)\}$ .

The possibilistic semantics introduced in [10] was defined for the family of possibilistic disjunctive logic programs. This means that the possibilistic clauses could have a disjunction in their heads. Since the possibilistic pstable semantics is defined for possibilistic normal programs, one can think that the possibilistic pstable semantics is less expressive than the possibilistic answer semantics. However, an interesting result is that the possibilistic pstable semantics is the same expressive than the possibilistic answer sets. By lack of space, we will omit the formal presentation of this result.

## 5 Conclusions

Uncertain information is present in many real applications *e.g.*, medical domain, weather forecast, *etc.* To find a suitable representation of this kind of information has been subject of much debate. The most common approaches for leading with this information are based on probability however some times; it is difficult to find suitable probabilities about some events.

According to the experimental side of Artificial Intelligence, it seem that a good form of capturing uncertain information is to adopted models where they could be close to the common sense of an expert of an area. For instance, MYCIN project [2] used estimates provided by expert physicians that reflect the tendency of a piece of evidence to prove or disprove a hypothesis.

Possibilistic logic is a type of logic of uncertainty tailored for reasoning under incomplete evidence and partially inconsistent knowledge. At the syntactic level it handles formulae of propositional or first-order logic to which are attached degrees of necessity. The degree of necessity (or certainty) of a formula expresses to what extent the available evidence entails the truth of this formula [3]. We argue that possibilistic logic is an excellent approximation of the approach adopted by MYCIN system which showed that it is practical in real applications.

In this paper, we introduce a new possibilistic logic programming semantics called possibilistic pstable semantics. This semantics is closer to possibilistic logic than the two possibilistic semantics defined until now [9,10]. Since the possibilistic pstable semantics is less syntactic sensitive than the possibilistic stable semantics [9] and the possibilistic answer set semantics [10], this semantics guaranties the existences of possibilistic pstable models. It is worth mentioning that since the possibilistic pstable semantics is based on pstable semantics which emerges from the fusion of paraconsistent logics and ASP, the possibilistic pstable semantics is influenced by paraconsistent logic and ASP.

## Acknowledgement

We are grateful to anonymous referees for their useful comments. J.C. Nieves wants to thank CONACyT for his PhD Grant.

## References

1. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press, Cambridge (2003)
2. Buchanan, B.G., Shortliffe, E.H.: Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley, London (1985)
3. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: Gabbay, D., Hogger, C.J., Robinson, J.A. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming. Nonmonotonic Reasoning and Uncertain Reasoning, vol. 3, pp. 439–513. Oxford University Press, Oxford (1994)
4. Halpern, J.Y.: Reasoning about uncertainty. MIT Press, Cambridge (2005)
5. Jakobovits, H., Vermeir, D.: Robust semantics for argumentation frameworks. *Journal of logic and computation* 9(2), 215–261 (1999)
6. McCarthy, J.: Some Expert System Need Common Sense. *Annals of the New York Academy of Sciences* 426(1), 129–137 (1984)
7. McCarthy, J., Hayes, P.J.: Some philosophical problems from the standpoint of artificial intelligence. In: Meltzer, B., Michie, D. (eds.) *Machine Intelligence* 4, pp. 463–502. Edinburgh University Press (1969) (reprinted in McC90)
8. Mendelson, E.: *Introduction to Mathematical Logic*, 4th edn. Chapman and Hall/CRC (1997)
9. Nicolas, P., Garcia, L., Stéphan, I., Lafèvre, C.: Possibilistic Uncertainty Handling for Answer Set Programming. *Annals of Mathematics and Artificial Intelligence* 47(1-2), 139–181 (2006)
10. Nieves, J.C., Osorio, M., Cortés, U.: Semantics for possibilistic disjunctive programs (poster). In: Chitta Baral, G.B., Schlipf, J. (eds.) *LPNMR 2007. LNCS (LNAI)*, vol. 4483, pp. 315–320. Springer, Heidelberg (2007)
11. Osorio, M., Navarro, J.A., Arrazola, J.R., Borja, V.: Ground nonmonotonic modal logic s5: New results. *Journal of Logic and Computation* 15(5), 787–813 (2005)
12. Osorio, M., Navarro, J.A., Arrazola, J.R., Borja, V.: Logics with Common Weak Completions. *Journal of Logic and Computation* 16(6), 867–890 (2006)

# Improving Efficiency of Prolog Programs by Fully Automated Unfold/Fold Transformation

Jiří Vyskočil and Petr Štěpánek

Department of Theoretical Computer Science and Mathematical Logic  
Charles University  
Malostranské náměstí 25  
118 00 Praha 1, Czech Republic  
petr.stepanek@mff.cuni.cz, jiri.vyskocil@mff.cuni.cz

**Abstract.** This paper is a contribution to improving computational efficiency of definite Prolog programs using Unfold/Fold (U/F) strategy with homeomorphic embedding as a control heuristic. Unfold/Fold strategy is an alternative to so called conjunctive partial deduction (CPD). The ECCE system is one of the best system for program transformations based on CPD. In this paper is presented a new fully automated system of program transformations based on U/F strategy. The experimental results, namely CPU times, the number of inferences, and the size of the transformed programs are included. These results are compared to the ECCE system and indicate that in many cases both systems have produced programs with similar or complementary efficiency.

Moreover, a new method based on a simple combination of both systems is presented. This combination represents, to our best knowledge, the most effective transformation program for definite logic programs. In most cases, the combination significantly exceeds both the Unfold/Fold algorithm presented here and the results of the ECCE system. The experimental results with a complete comparison among these algorithms are included.

**Keywords:** logic programming, prolog, partial deduction, unfold/fold transformation, homeomorphic embedding.

## 1 Introduction

In the present paper, we describe a transformation motivated by Unfold-Definition-Fold method (UDF) due to [8] and by partial deduction. It terminates for all definite logic programs and is fully automated [1]. Its termination proof is obtained using homeomorphic embedding used first by Dershowitz [2]

---

<sup>1</sup> It means that the transformation algorithm presented in this paper terminates and it is applicable to all definite logic programs (which are given as input). It does not mean that the algorithm converts non-terminating programs into terminating ones.

and some techniques similar to those used in the termination proof of the conjunctive partial deduction [3]. In most cases, the transformed programs have better computational behaviour than the original ones. Experimental results measuring time and a number of inferences are included. The results obtained are compared with the results of conjunctive partial deduction implemented in the ECCE system. Although the present method is not optimized and has no pre- and only very weak post-processing, the results are encouraging.

We suppose, that the reader is well acquainted with the standard of concept of unfolding and folding. In the paper, we shall adopt the notation used in Apt [1].

## 2 Eliminating Unnecessary Variables – Motivation

Existential variables are often used in logic programs for storing intermediate results and multiple variables are used for multiple structure traversals. Thus in many cases, eliminating them improves the efficiency of the program.

**Definition 1.** (*Unnecessary variables*)

Given a clause  $c$ , and a program  $P$

1. We say that variables occurring only in the body of  $c$  are *existential* variables.
2. Variables that occur in the body of  $c$  more than once are called *multiple* variables.
3. Both *existential* and *multiple* variables in the clauses of  $P$  are called *unnecessary variables* of  $P$ .

As shown in [8] there are two classes of definite programs for which all unnecessary variables can be eliminated by repeating UDF (unfolding-definition-folding) transformation steps in this order. The authors of [8] showed that in general it is undecidable whether, for any input, this elimination procedure terminates<sup>2</sup>.

We present a new transformation that is motivated by the above UDF method which terminates for all definite Prolog programs. However, it may not eliminate all unnecessary variables. The algorithm has been implemented as fully automated and in most cases the resulting programs are computationally more efficient than the original ones.

First we illustrate the transformation process on an example.

### *Example 1*

Consider a naive but intuitive version of DOUBLEAPPEND used in [3] which concatenates three lists.

<sup>2</sup> The authors of [8] also defined so called extended elimination procedure which terminates for all definite programs but may not eliminate all unnecessary variables. Our algorithm uses a different control and termination mechanism than it is proposed in [8].

- (1) `doubleapp(X,Y,Z,XYZ) :- append(X,Y,XY),append(XY,Z,XYZ).`
- (2) `append([],L,L).`
- (3) `append([H|X],Y,[H|Z]) :- append(X,Y,Z).`

In the first clause, there are two occurrences of the unnecessary variable `XY`. Using the strategy `Unfold-Definition-Fold`, we remove this variable and obtain a less intuitive but more efficient version of `DOUBLEAPPEND`.

At the beginning, we unfold the clause (1) w.r.t. `append(X,Y,XY)` and replace the clause (1) by the clauses (1a) and (1b).

- (1a) `doubleapp([],Y,Z,XYZ) :- append(Y,Z,XYZ).`
- (1b) `doubleapp([H|X],Y,Z,XYZ) :- append(X,Y,XY),append([H|XY],Z,XYZ).`
- (2) `append([],L,L).`
- (3) `append([H|X],Y,[H|Z]) :- append(X,Y,Z).`

Then we define a new clause (4) for further transformation.

- (4) `newp1(H,X,Y,Z,XYZ) :- append(X,Y,XY),append([H|XY],Z,XYZ).`

Then fold the body of the clause (1b) with the definition clauses (4) and get the clause (1f).

- (1a) `doubleapp([],Y,Z,XYZ) :- append(Y,Z,XYZ).`
- (1f) `doubleapp([H|X],Y,Z,XYZ) :- newp1(H,X,Y,Z,XYZ).`
- (2) `append([],L,L).`
- (3) `append([H|X],Y,[H|Z]) :- append(X,Y,Z).`
- (4) `newp1(H,X,Y,Z,XYZ) :- append(X,Y,XY),append([H|XY],Z,XYZ).`

We continue with unfolding of the definition clause (4) w.r.t. `append([H|XY],Z,XYZ)`. We replace the clause (4) by the clause (4u).

- (4u) `newp1(H,X,Y,Z,[H|XYZ]) :- append(X,Y,XY),append(XY,Z,XYZ).`

Then we fold the clause (4u) with the original definition of `doubleapp/4` (1) and get the clause (4f).

- (4f) `newp1(H,X,Y,Z,[H|XYZ]) :- doubleapp(X,Y,Z,XYZ).`

We continue with unfolding the clause (1f) and we get the clause (1u).

- (1u) `doubleapp([H|X],Y,Z,[H|XYZ]) :- doubleapp(X,Y,Z,XYZ).`

Finally, the transformation is complete. We get a more efficient version of the program without double traversing of the intermediate list. The transformed program reads as follows.

- (1a) `doubleapp([],Y,Z,XYZ) :- append(Y,Z,XYZ).`
- (1u) `doubleapp([H|X],Y,Z,[H|XYZ]) :- doubleapp(X,Y,Z,XYZ).`
- (2) `append([],L,L).`
- (3) `append([H|X],Y,[H|Z]) :- append(X,Y,Z).`

The above example shows that we need an instrument for the choice of atoms for unfolding (it was the first and the second unfolding step in our demonstration) and another instrument for proving the termination of the algorithm. To solve the latter requirement, we use a version of homeomorphic embedding, [2,3,5] which was used for a termination proof of Conjunctive partial deduction. To solve the former one we use a syntactic method.

### 3 Homeomorphic Embedding, Linking Variables and Safe Selection

The idea of homeomorphic embedding on sets of terms goes back to Kruskal [4] and Nash-Williams [7].

**Definition 2.** (*The homeomorphic embedding*)

Assume that we have a language with finite special symbols and the potentially infinite set of variables. Note that every successful computation uses only finitely many variables.

The homeomorphic embedding  $\sqsubseteq$  on the set of all atomic formulas and terms is defined inductively as follows:

- $X \sqsubseteq Y$  for all variables,
- (diving rule)  $s \sqsubseteq f(t_1, \dots, t_n)$  if  $s \sqsubseteq t_i$  for some  $i \in \{1 \dots n\}$ ,
- (coupling rule)  $f(s_1, \dots, s_n) \sqsubseteq f(t_1, \dots, t_n)$  if  $s_i \sqsubseteq t_i$  for every  $i \in \{1 \dots n\}$ ,

where  $s, s_i, t_i$  are terms, and in diving and coupling rule  $n = 0$  is allowed.

Intuitively, an expression  $A$  is homeomorphically embedded into an expression  $B$  iff  $B$  is more complex than  $A$  in the following sense:  $A$  can be obtained from  $B$  by “simplifying” some subexpressions of  $B$ .

Note that all variables are treated in the same way.

It was shown by Kruskal [4] and Nash-Williams [7] that the relation  $\sqsubseteq$  of homeomorphic embedding is a well quasi-ordering on the set of all terms and atomic formulas. De Schreye et al. [3] used this result to prove termination of their algorithm of conjunctive partial deduction. We use a similar approach (but not the same) in proving termination of our Unfold/Fold algorithm.

**Definition 3.** (*Partition of the body of a clause*)

Given a set of atoms, we define a binary relation  $\sim$  as follows. For two atoms  $A, B$ , we put

$$A \sim B \text{ iff } Vars(A) \cap Vars(B) \neq \emptyset$$

It means that  $A \sim B$  iff  $A$  and  $B$  have at least one variable in common. Obviously,  $\sim$  is symmetric and reflexive, thus the transitive closure  $\approx$  of  $\sim$  is an equivalence relation.



Given a clause  $c$

$$H \leftarrow A_1, \dots, A_n$$

We denote by  $PartB(c)$  [\[3\]](#) the partition of the set  $\{A_1, \dots, A_n\}$  of the atoms of the body of  $c$  to disjoint subsets induced by the equivalence relation  $\approx$ . The elements of the partition are called the *blocks of the clause*  $c$ .

**Definition 4.** (*Linking variables of a block*)

Let  $c$  be a clause  $H \leftarrow A_1, \dots, A_n$  and let  $B \in PartB(c)$  be a block in the body of  $c$ . The set of the *linking variables* of  $B$  is defined as follows

$$LinkVars_c(B) = Vars(H) \cap Vars(B)$$

Note that linking variables are not existential variables of the clause  $c$ .

**Definition 5.** (*Faithful variant of a block*)

A Block  $B_1$  of a clause  $c_1$  is called a *faithful variant* of a block  $B_2$  of a clause  $c_2$  iff there exists a renaming substitution  $\theta$  such that the following conditions hold

- (i)  $B_1 = B_2\theta$  ( $B_1$  is a variant of  $B_2$ )
- (ii)  $(\forall X \in Vars(B_2)) (X \in LinkVars_{c_2}(B_2) \leftrightarrow X\theta \in LinkVars_{c_1}(B_1))$

**Definition 6.** (*Safe selected atom*)

Let  $\triangleleft$  be the homeomorphic embedding on the set of atoms and terms defined in Definition [\[2\]](#). Let  $c$  be a clause

$$H \leftarrow A_1, \dots, A_k, \dots, A_n$$

which arises from an original clause by sequential unfolding via selected atoms  $B_1, \dots, B_m$ . We say that the atom  $A_k$  is *safe selected for unfolding* iff  $A_k \not\triangleleft B_i$  for each atom  $B_i$ .

## 4 The Algorithm

The algorithm is motivated by the Unfold-Definition-Fold strategy presented in [\[8\]](#). The key idea in this new algorithm is the selection of a clause and of one of its atoms for Unfolding. The selection rule used in the presented algorithm makes the transformation applicable to all definite logic programs. In most cases, the transformed programs are more effective than original ones (see Section [\[5\]](#)). The transformation is fully automated.

Our algorithm improves the original elimination procedure [\[8\]](#) which limits the class of programs for which the algorithm is applicable and has no fully automated implementation. The complete elimination of unnecessary variables

---

<sup>3</sup> The  $PartB(c)$  is implemented in our algorithm as follows. An input is a list of atoms with an ordering from the original clause  $c$ . An output is a list of lists, where each element from the output list corresponds to each block from  $PartB(c)$  with a preserved ordering of atoms from the original clause  $c$  inside each block.

**Algorithm 1.** Procedure for elimination of unnecessary variables

Input: A definite logic program  $P$ .

Output: A set  $\mathbf{Transf}P$  of clauses such that for each goal  $G$  consisting of predicates of  $P$ , the answer set of  $P$  via  $G$  is equal to answer set of  $\mathbf{Transf}P$  via  $G$  (as shown in Example 11 the set of clauses  $\mathbf{Transf}P$  can be interpreted as the resulting transformed program).

1. let  $Q$  be an empty queue of pairs  $\langle \text{Clause}, \text{History} \rangle$  where  $\text{History}$  is a set of atoms and  $\text{Clause}$  is a definite clause;
2.  $\mathbf{Transf}Cs, D, Cs := \emptyset$ ;
3. for each clause  $C \in P$ :
  - (i) if  $C$  contains at least one unnecessary variable, then  
 $Cs := Cs \cup \{C\}$ ;
  - (ii) if the head in  $C$  occurs only once as the head in  $P$ , then  
 $D := D \cup \{C\}$ ;
4. for each clause  $C \in Cs$  push pair  $\langle C, \emptyset \rangle$  into  $Q$ ;
5. while  $Q \neq \emptyset$  do

**Unfolding steps:**

  - $\langle C, H \rangle := \text{pop } Q$ ;
  - select the first *safe selected atom*  $A$  from the body of  $C$ , that is:
    - for each  $h \in H$   $A \not\prec h$ ;
    - if no such  $A$  exists then  
 $\mathbf{Transf}Cs := \mathbf{Transf}Cs \cup \{C\}$ ; goto 5;
    - else unfold atom  $A$  in the body of  $C$  using clauses in  $P$  and store resulting unfolded clauses in the set  $Us$ ;

**Definition steps:**

  - for each clause  $E \in Us$ 
    - for each block  $B \in \text{PartB}(E)$  such that:
      - (i)  $B$  contains at least one unnecessary variable, and
      - (ii)  $B$  is not a *faithful variant* of the body of any clause which is in  $D$  then  
 $\text{let } F := \text{newp}(X_1, \dots, X_n) :- B$ ,  
 where  $\text{newp}$  is a fresh predicate symbol and  $X_1, \dots, X_n$  are  
*linking variables* of  $B$  with respect to the head of  $E$ ,  
 $\text{push}$  the pair  $\langle F, H \cup \{A\} \rangle$  into  $Q$ ,  
 $D := D \cup \{F\}$ ;

**Folding steps:**

  - for each clause  $E \in Us$ 
    - begin
      - for each block  $B \in \text{PartB}(E)$  such that:  
 $B$  is a *faithful variant* of body of a clause  $N$  from  $D$ ,  
 then fold  $B$  in  $E$  using  $N$  to obtain  $E$ ;

$\mathbf{Transf}Cs := \mathbf{Transf}Cs \cup \{E\}$

end;
6. for each clause  $E \in \mathbf{Transf}Cs$  such that:
  - $E$  contains atoms  $As$  in its body which are defined by at most one clause from  $(P \setminus Cs) \cup \mathbf{Transf}Cs$  (it means that there is at most one clause in  $(P \setminus Cs) \cup \mathbf{Transf}Cs$  with the head which is unifiable with an atom in the body of  $E$ ),
  - then replace  $E$  in  $\mathbf{Transf}Cs$  by  $E$  with all atoms in  $As$  unfolded;
7.  $\mathbf{Transf}P := (P \setminus Cs) \cup \mathbf{Transf}Cs$ ;
8. end.

in the algorithm is not guaranteed for all definite logic program (more about necessary conditions for elimination can be found in [8]).

The termination proof uses the concepts of safe selection and homeomorphic embedding. Numbers 1 - 5 are the main part of the algorithm. Number 6 is a post-processing part. Proofs of termination and correctness of the presented algorithm are beyond the scope of this paper and can be found in [11].

## 5 Benchmarks

The following table compares several comparative benchmarks for original Prolog sources<sup>4</sup>, sources transformed by ECCE 1.1 [6], sources transformed by Algorithm [7] and sources transformed by a combination of both algorithms. The comparison was made on one representative predefined goal for each tested program. The original Prolog program is denoted by  $P$  in the input for Algorithm 1 (UDF).

To make a comparison of the above algorithm we put to ECCE the original Prolog source  $P$  and a specialisation goal for partial evaluation. This goal is a predicate from the predefined test goal with free variables for all its arguments. The following flags were enabled in ECCE: `RAF Filtering`, `FAR Filtering`, `Dead Code Elimination`, `Remove Redundant Calls`, `Determinate Post Unfolding`, `Reduce Polyvariants`. The combination of both algorithms works in the following way. First, the original source was transformed by Algorithm 1 (UDF) and then the output was put as an input for ECCE.

Because the ECCE system is nowadays probably the best fully automated partial evaluation system based on Conjunctive Partial Deduction (CPD) it is very interesting to compare it with our system which is to the best of our knowledge the first fully automated algorithm based on Unfold/Fold transformations.

Measurements of the number of inferences<sup>6</sup>, of the inference speedup and of the number of clauses have been performed on SWI-Prolog 5.2.13. These measured parameters (with the exception of time) do not depend on any specific operating system or hardware architecture. Measurements of CPU runtime and CPU runtime speedup have been performed on SWI-Prolog 5.2.13 with default settings (without any options). The testing machine had following parameters: Processor: Intel Pentium 4 - 3.4GHz, RAM: 2 GB, Operating system: Debian GNU/Linux system with kernel 2.6.14.3. The results are shown in the following table<sup>7</sup>.

<sup>4</sup> In the table this program is marked as none transformation.

<sup>5</sup> In the table this program is marked as UDF (Unfold-Definition-Fold) transformation.

<sup>6</sup> The exact definition of number of inferences can be found in the manual of SWI-Prolog. It roughly corresponds to the number of resolution steps.

<sup>7</sup> Complete source codes, test goals and more benchmarks on various machines and various Prolog systems can be found at <http://kti.mff.cuni.cz/~vyskocil/research/MICAI07/>



A short description of benchmark programs follows<sup>8</sup>:

- rotate\_leftdepth:** Computes possible rotations of a binary tree that give a path to a leftmost leaf of the same length.
- frontier:** Gives a list of the leaves of a binary tree.
- Ackermann:** Computes the Ackermann function.
- inorder:** Collects all node values in a binary tree into a list.
- rotate:** Computes a very slow double rotation of a binary tree.
- tree\_sum\_size:** Computes a sum of a tree size.
- permutation:** Computes all permutations of a list.
- fibb:** Computes the Fibonacci numbers.
- doubleapp:**<sup>9</sup> Appends 3 lists into 1 (it is the same program as defined in Section 3).
- applast:**<sup>9</sup> Appends an element to a list and after that extracts the last element from the list.
- advisor:**<sup>9</sup> A simple Prolog program as a weather advisor.
- depth:**<sup>9</sup> Computes the depth of a Prolog goal.
- ex\_depth:**<sup>9</sup> A variant of depth with meta-interpretation.
- goat:**<sup>9</sup> Computes the well known Wolf-Goat-Cabbage problem.
- flip:**<sup>9</sup> Flips a tree structure twice (thus returns back the original tree).
- grammar:**<sup>9</sup> It is a parser of a simple grammar.
- model\_elim:**<sup>9</sup> Computes a simple model elimination.
- transpose:**<sup>9</sup> Transposes matrices (of any dimension).
- relative:**<sup>9</sup> A simple program which represents family relations.

As it can be seen from the table, the speed (inference speedup and CPU runtime speedup) of UDF on the tested program increased after transformation in most cases. In many cases the speed was almost the same and in only one case the speed of UDF was considerably lower than that of the original Prolog program. The average inference speedup on tested programs was approximately 69% (ECCE had 67%). Average CPU runtime speedup was approximately 31% (ECCE had 72%).

On one hand the measurement of CPU runtime is not ideal because the measured speedup can differ on various machines with the same Prolog system and the same operating system up to  $\pm 10\%$ <sup>7</sup>. When a different Prolog system is used, the situation is even worse. But on the other hand, such measurements represent the real run of the tested program while the number of inferences is a more theoretical measure.

As it can be seen from the table, there are many cases where both methods ECCE and UDF were comparable (small negative differences of UDF are caused by a much better post-processing of ECCE). But there are some cases where

<sup>8</sup> Complete source codes, test goals and more benchmarks on various machines and various Prolog systems can be found at <http://kti.mff.cuni.cz/~vyskocil/research/MICAI07/>

<sup>9</sup> This program is from the DPPD (Dozens of Problems for Partial Deduction) library at <http://www.ecs.soton.ac.uk/~mal/systems/dppd.html>

UDF method was better and vice versa. The UDF was significantly better than ECCE in seven cases, significantly worse than ECCE in seven cases and almost equivalent with ECCE in five cases. These experimental results may indicate that both methods are orthogonal.

The logical conclusion of the previous results was to combine both methods together. The result is very interesting and it shows that the combination of both methods produces almost always significantly better results (in comparison with previous two methods). The average inference speedup of the combined method on tested programs was approximately 136% and average CPU runtime speedup was approximately 111% (both numbers are compared with the original Prolog programs).

The size of programs transformed by UDF did not increase more than 6 times, the average code size coefficient of all tested programs was approximately 2.4 times in comparison with the original Prolog programs. The ECCE had in most cases less number of clauses because of its better post-processing such as **Dead Code Elimination**. The combination did not increase the code size more than 11 times.

We have to say that it was very surprising for us that the first attempt of fully automated Unfold/Fold transformation method is quite comparable to the most advanced CPD ECCE system. Moreover, more than one half of test programs belonged to the DPPD library which was originally designed for testing ECCE, and even for these examples UDF had quite comparable results. The combination of both methods gave significantly better results than each method applied individually. Moreover, this combination represents, to our best knowledge, the most effective transformation program for definite logic programs.

## 6 Conclusions and Future Work

In the paper we have described a fully automated algorithm for eliminating unnecessary variables in definite programs. We have implemented this algorithm and applied it to several benchmarks of programs. The results showed that the transformed programs have approximately 31% speedup (measured in CPU runtime) compared to non-transformed originals. But there still exist programs for which UDF has worse results after transformation than their originals.

In the future work we would like to identify the class of programs that have worse results and improve the behaviour of the algorithm on these programs. However, the experiments have shown that Unfold/Fold is a powerful strategy that is in some sense complementary to conjunctive partial deduction strategy. We believe that this method has a comparable asymptotic time complexity as CPD presented in [3,5] since we used a similar Homeomorphic Embedding termination control mechanism. To describe the exact time complexity is out of the scope of this paper.

Surprisingly, a new algorithm that simply combines UDF and CPD methods produces significantly better results than each of them. The average CPU

runtime speedup of the combined method is 111% and this algorithm seems to be the most effective transformation algorithm for definite logic programs.

As the described algorithm is fully automated, it would be valuable to extend it to all general logic programs (and possibly to full Prolog).

## References

1. Apt, K.R.: From Logic Programming to Prolog. Prentice-Hall, Englewood Cliffs (1996)
2. Dershowitz, N.: Termination in rewriting. *Journal of Symbolic Computation* 3, 69–116 (1987)
3. De Schreye, D., Glück, R., Jørgensen, J., Leuschel, M., Martens, B., Sørensen, M.H.: Conjunctive partial deduction: foundations, control, algorithms and experiments. *The Journal of Logic Programming* 41, 231–277 (1999)
4. Kruskal, J.B.: Well-quasi-ordering, the Tree Theorem, and Varsonyi's conjecture. *Trans. Amer. Math. Society* 95, 210–225 (1960)
5. Leuschel, M.: Improving Homeomorphic Embeddings for On line Termination. In: Flener, P. (ed.) LOPSTR 1998. LNCS, vol. 1559, pp. 199–218. Springer, Heidelberg (1999)
6. Leuschel, M.: Ecce partial deduction system, <http://www.ecs.soton.ac.uk/~mal/systems/ecce.html>
7. Nash-Williams, C.: On well quasi ordering finite trees. *Proc. Camb. Phil. Society* 59, 833–835 (1963)
8. Proietti, M., Pettorossi, A.: Unfolding-definition-folding, in this order, for avoiding unnecessary variables in logic programs. *Theoretical Computer Science* 142, 89–124 (1995)
9. Tamaki, H., Sato, T.: Unfold/Fold Transformation of Logic Programs. In: Tärnlund, S. (ed.) ICLP 1984, pp. 127–138, Uppsala University, Sweden (1984)
10. Turchin, V.F.: Program transformation with metasystems transitions. *Journal of Functional Programming* 3(3), 283–313 (1993)
11. Vyskočil, J., Štěpánek, P., Halama, M.: Speedup by Fully Automated Unfold/Fold Transformation. Technical Report TR No 2006 /1 Department of KTIML MFF, Charles University in Prague (2006)

# A Word Equation Solver Based on Levensthein Distance

César L. Alonso<sup>1,\*</sup>, David Alonso<sup>1</sup>, Mar Callau<sup>2</sup>, and José Luis Montaña<sup>3,\*\*</sup>

<sup>1</sup> Centro de Inteligencia Artificial, Universidad de Oviedo  
Campus de Viesques, 33271 Gijón, Spain  
calonso@aic.uniovi.es

<sup>2</sup> Universidad Complutense de Madrid  
marzori@gmail.com

<sup>3</sup> Departamento de Matemáticas, Estadística y Computación,  
Universidad de Cantabria  
montanjl@unican.es

**Abstract.** Many regularity properties of strings, like those appearing in hardware specification and verification, can be expressed in terms of word equations. The solvability problem of word equations is NP-hard and the first algorithm to find a solution for a word equation, when this solution exists, was given by Makanin in 1977. The time complexity of Makanin's algorithm is triple exponential in the length of the equations. In this paper we present an evolutionary algorithm with a local search procedure that is efficient for solving word equation systems. The fitness function of our algorithm is based on Levensthein distance considered as metric for the set of 0-1 binary strings. Our experimental results evidence that this metric is better suited for solving word equations than other edit metrics like, for instance, Hamming distance.

**Keywords:** Word equations, evolutionary computation, local search strategies.

## 1 Introduction

In 1977, Makanin proved that the satisfiability problem for word equation systems over free semigroups is decidable [1]. Makanin reached this result by presenting a very complicated algorithm which solves the satisfiability problem for word equations with constants. The algorithm is based on the construction of a finite search graph. The finiteness proof of the search graph is among the most complex proofs in theoretical computer science. The nondeterministic time complexity of the algorithm is triple exponential and the problem remains NP-hard (see [3]).

The problem of solving equations in algebras is a well-established area in computer science with a wide range of applications ([4]). Efficient automation of word equations reasoning is essential not only to deal with problems having a theoretical flavor like characterization of imprimitiveness, string unification in PROLOG-3 or unification in theories with associative non-commutative operators, but also in some industrial problems that require the effective mechanization of many hardware verification proofs. As

---

\* Partially supported by the Spanish grant MTM2004-01176.

\*\* Partially supported by the Spanish grant TIN2007-67466-C02-02.



example, it has been demonstrated that the lack of such capability forms one of the main impediments to cost-effective verification of some industrial-sized microprocessors (see [17]).

After Makanin’s algorithm, better complexity upper bounds have been obtained: EXPSPACE ([8]), NEXPTIME ([15]) and PSPACE ([13]). Methods for solving particular instances of the problem, as the case where each variable appears at most twice in the equation ([16]), or studies about the space complexity of the problem over free groups ([9]) were also proposed.

Recently, the Word Equation System (WES) problem has been studied in the framework of Evolutionary Computation. Up to our knowledge the first algorithms for WES instances by means of genetic algorithms are given in [1] and [2].

In the present paper we propose a new evolutionary algorithm with a local search procedure for solving the WES problem. The individuals are codified as 0-1 binary strings with variable length and suitable recombination operators are defined. For the fitness function we use the Levenstein distance between strings. Two versions of an appropriate local search procedure for the Levenstein distance are defined: one of them explores all possibilities for each gene and the other explores only a randomly chosen one. In a first approach an upper bound for the length of the solutions is given to the algorithm, but we also treat the problem in its more general form without bounds. The paper is organized as follows: Section 2 contains the basic concepts related with WES problem; Section 3 describes the main components of our evolutionary algorithm and the local search procedures; Section 4 includes the experimental results after solving some randomly generated WES instances; finally, Section 5 contains some conclusive remarks.

## 2 Statement of the Problem

Let  $A$  be a finite alphabet of constants and let  $\Omega$  be an alphabet of variables. We assume that these alphabets are disjoint. As usual we denote by  $A^*$  the set of words on  $A$ ;  $|w|$  denotes the length of  $w \in A^*$  and  $\varepsilon$  stands for the empty word.

**Definition 1.** *A word equation over  $A$  with variables in  $\Omega$  is a pair  $E = (L, R) \in (A \cup \Omega)^* \times (A \cup \Omega)^*$ , denoted by  $L = R$ . A word equation system (WES) over  $A$  with variables in  $\Omega$  is a finite set  $S = \{E_1, \dots, E_n\}$ , where  $E_i = (L_i, R_i)$  is a word equation over  $A$  with variables in  $\Omega$  for all  $i \in \{1, \dots, n\}$ . The length of a word equation  $E = (L, R)$  is  $|E| := |L| + |R|$ .*

**Definition 2.** *Let  $S = \{E_1, \dots, E_n\}$ ,  $E_i = (L_i, R_i)$ ,  $i \in \{1, \dots, n\}$ , be a WES over  $A$  with variables in  $\Omega$ . A solution of  $S$  is a morphism  $\sigma : (A \cup \Omega)^* \rightarrow A^*$  such that  $\sigma(a) = a$ , for all  $a \in A$ , and  $\sigma(L_i) = \sigma(R_i)$ , for all  $i \in \{1, \dots, n\}$ .*

**Definition 3.** *Let  $S$  be a WES over  $A$  with variables in  $\Omega$  as in the previous definition, and a solution  $\sigma$  of  $S$ . The length of  $\sigma$  is defined as follows:*

$$|\sigma| = \max\{|\sigma(x)| : x \in \Omega\}$$

The WES problem can be stated as follows: given a word equation system  $S$  as input, find a solution of  $S$  if there exists anyone or determine the no existence of solutions otherwise. Makanin gave an upper bound for the minimal length of a solution of  $S$ . This upper bound is double exponential in the sum of the lengths of the equations in the system. Nevertheless it is generally believed a conjecture that reduces to single exponential this upper bound. Although this conjecture was true, the search space guaranteeing to contain a solution of  $S$ , if there exists one, is extremely large. A simpler problem than the stated above is the  $d$ -WES problem.

*d-WES problem:* Given a WES  $S$  over  $A$  with variables in  $\Omega$ , find a solution  $\sigma : (A \cup \Omega)^* \rightarrow A^*$  such that  $|\sigma| \leq d$  or determine the no existence otherwise.

*Example 4.* (see [1]) For each  $n \geq 1$ , let  $F_n$  and  $WordFib_n$  be the  $n$ -th Fibonacci number and the  $n$ -th Fibonacci word over the alphabet  $A = \{0, 1\}$ , respectively. For any  $n \geq 2$  let  $S_n$  be the word equation system over the alphabet  $A = \{0, 1\}$  and variables set  $\Omega = \{x_1, \dots, x_{n+1}\}$  defined as:

$$\begin{aligned} x_1 &= 0 \\ x_2 &= 1 \\ 01x_1x_2 &= x_1x_2x_3 \\ &\dots \\ 01x_1x_2x_2x_3 \dots x_{n-1}x_n &= x_1x_2x_3 \dots x_{n+1}. \end{aligned}$$

Then, for any  $n \geq 2$ , the morphism  $\sigma_n : (A \cup \Omega)^* \rightarrow A^*$ , defined by

$$\sigma_n(x_i) = FibWord_i,$$

for  $i \in \{1, \dots, n + 1\}$ , is the only solution of the system  $S_n$ . This solution satisfies  $|\sigma_n(x_i)| = F_i$ , for each  $i \in \{1, \dots, d + 1\}$  and hence  $|\sigma_n| = F_{n+1}$ . Recall that  $FibWord_1 = 0$ ,  $FibWord_2 = 1$  and  $FibWord_i = FibWord_{i-2}FibWord_{i-1}$  if  $i > 2$ .

The above example shows that any algorithm which solves WES problem in its general form must have, at least, exponential worst-case complexity if words are codified as strings. This is due to the fact that the system  $S_n$  has polynomial size in  $n$  and the only solution of  $S_n$ , namely  $\sigma_n$ , has exponential length with respect to  $n$ , because it contains the  $n$ -th Fibonacci word,  $WordFib_n$ . Note that  $WordFib_n$  has length equal to the  $n$ -th Fibonacci number,  $F_n$ , which is exponential w.r.t  $n$ . This exponential length argument cannot be exhibited for a lower complexity bound of the  $d$ -WES problem. But  $d$ -WES problem remains NP-complete for any  $d \geq 2$  because the 3-SAT problem is polynomially reducible to 2-WES problem (see [1], [8], [9] for the details).

### 3 The Local Search Genetic Algorithm

Given an alphabet,  $A$ , and some string over  $A$ ,  $\alpha \in A^*$ , for any pair of positions  $i, j$ ,  $1 \leq i \leq j \leq |\alpha|$ , in the string  $\alpha$ ,  $\alpha[i, j] \in A^*$  denotes the substring of  $\alpha$  given by the extraction of  $j - i + 1$  consecutive many letters  $i$  through  $j$  from string  $\alpha$ . In the case  $i = j$ , we denote by  $\alpha[i]$  the single letter substring  $\alpha[i, i]$ , which represents the  $i$ -th symbol of string  $\alpha$ .

#### 3.1 Representation of the Individuals

Along these pages we restrict to the alphabet  $A = \{0, 1\}$ . Let  $S$  be a word equation system over  $A$  and set of variables  $\Omega = \{x_1, \dots, x_m\}$ . A solution  $\sigma$  of  $S$  will be represented by a chromosome  $\bar{\alpha}$  which consists of a list of  $m$  strings  $(\alpha_1, \dots, \alpha_m)$ ,  $\alpha_i \in A^*$ , for all  $i \in \{1, \dots, m\}$ . String  $\alpha_i$  represents the value of the variable  $x_i$ , i.e.  $\alpha_i = \sigma(x_i)$ . The length of chromosome  $\bar{\alpha}$  is defined as  $|\bar{\alpha}| = \sum_{i=1}^m |\alpha_i|$ .

#### 3.2 Fitness Function

First, we consider a notion of distance between strings which extends Hamming distance (valid only for equal size strings). This is necessary because the chromosomes (representing candidate solutions for our problem instances) are variable size strings.

To this end we propose Levenshtein distance (see [10]), which is a measure of similarity between two strings, often used in spelling correction, plagiarism detection, molecular biology and speech recognition.

The Levenshtein distance of two bit strings  $\alpha, \beta \in A^*$ ,  $LD(\alpha, \beta)$ , is defined as the minimum number of bit mutations required to change  $\alpha$  into  $\beta$ , where a bit mutation is one of the following operations: flip a bit, insert a new bit or delete a bit.

Function  $LD$  can be computed by a well known dynamical programming strategy based on a recursion where a two dimensional matrix  $M[0 \dots |\alpha|, 0 \dots |\beta|]$  is used to hold the  $LD$  values:

$$M[i, j] = LD(\alpha[1, i], \beta[1, j]) \tag{1}$$

and is defined as follows:

$$\begin{aligned} M[0, 0] &= 0 \\ M[i, 0] &= i, \quad i = 1, \dots, |\alpha| \\ M[0, j] &= j, \quad j = 1, \dots, |\beta| \\ M[i, j] &= \min\{M[i - 1, j - 1] + LD(\alpha[i], \beta[j]), M[i - 1, j] + 1 \\ &\quad M[i, j - 1] + 1\} \quad 1 \leq i \leq |\alpha|, 1 \leq j \leq |\beta| \end{aligned}$$

*Remark 5.* The matrix  $M$  can be computed row by row: the  $i$ -th row depends only on  $i - 1$ -th row. The time complexity of this algorithm is  $O(|\alpha||\beta|)$ . There are faster algorithms for the  $LD$  problem. Some of these algorithms are fast if certain conditions holds, e.g. the strings are similar, or dissimilar, or the alphabet is large,... Ukkonen has given an algorithm with worst case complexity  $O(nd)$  and average complexity  $O(n + d^2)$  where  $n$  is the length of the string and  $d$  is its  $LD$ . This is fast for similar strings where  $d$  is small, i.e.  $d \ll n$  (see [18]).

Given a word equation system  $S = \{L_1 = R_1, \dots, L_n = R_n\}$  over the alphabet  $A = \{0, 1\}$  with set variables  $\Omega = \{x_1, \dots, x_m\}$  and a chromosome  $\bar{\alpha} = (\alpha_1, \dots, \alpha_m)$ , representing a candidate solution for  $S$ , the fitness of  $\bar{\alpha}$  is computed as follows:

First, in each equation, we substitute, for  $j \in \{1, \dots, m\}$ , every variable  $x_j$  for the corresponding string  $\alpha_j \in A^*$ , and, after this replacement, we get the expressions  $\{L_1(\bar{\alpha}) = R_1(\bar{\alpha}), \dots, L_n(\bar{\alpha}) = R_n(\bar{\alpha})\}$  where  $\{L_i(\bar{\alpha}), R_i(\bar{\alpha})\} \subset A^*$  for all  $i \in \{1, \dots, n\}$ .

Then, the fitness of the chromosome  $\bar{\alpha}$ ,  $f(\bar{\alpha})$ , is defined as:

$$f(\bar{\alpha}) = \sum_{i=1}^n LD(L_i(\bar{\alpha}), R_i(\bar{\alpha})) \tag{2}$$

*Remark 6.* Let  $S = \{L_1 = R_1, \dots, L_n = R_n\}$  be a word equation system over the alphabet  $A = \{0, 1\}$  with set variables  $\Omega = \{x_1, \dots, x_m\}$  and let  $\bar{\alpha} = (\alpha_1, \dots, \alpha_m)$  be a chromosome representing a candidate solution for  $S$ . Define the morphism  $\sigma : (A \cup \Omega)^* \rightarrow A^*$  as  $\sigma(x_i) = \alpha_i$ , for each  $i \in \{1, \dots, m\}$ . Then the morphism  $\sigma$  is a solution of system  $S$  if and only if the fitness of the chromosome  $\bar{\alpha}$  is equal to zero, that is  $f(\bar{\alpha}) = 0$ .

### 3.3 Genetic Operators

**selection:** For the selection operator we make use of the roulette wheel selection procedure (see [6]). Let  $\bar{\alpha}_1, \dots, \bar{\alpha}_t$  be a population of chromosomes. Since our aim is to minimize the objective function  $f$  defined above, we perform the following transformation. Assume  $f(\bar{\alpha}_i) \neq 0$  for all  $i \in \{1, \dots, t\}$ . Let  $M$  the lowest common multiple of  $\{f(\bar{\alpha}_1), \dots, f(\bar{\alpha}_t)\}$ . Define  $f_{aux}(\bar{\alpha}_i) = \frac{M}{f(\bar{\alpha}_i)}$ . Now the probability of selection of chromosome  $\bar{\alpha}_i$  is:

$$\frac{f_{aux}(\bar{\alpha}_i)}{\sum_{j=1}^t f_{aux}(\bar{\alpha}_j)} \tag{3}$$

**crossover:** (see [2]) Given two chromosomes  $\bar{\alpha} = (\alpha_1, \dots, \alpha_m)$  and  $\bar{\beta} = (\beta_1, \dots, \beta_m)$ , the result of a crossover is a new chromosome  $\bar{\gamma} = (\gamma_1, \dots, \gamma_m)$  where  $\gamma_i$  is constructed applying a local crossover to the strings  $\alpha_i, \beta_i$ , for all  $i \in \{1, \dots, m\}$ . Assume  $l_1 = |\alpha_i| \leq |\beta_i| = l_2$ , then  $\gamma_i = \gamma_i[1, l_1]\beta_i[l_1 + 1, q]$ , where  $\gamma_i[1, l_1]$  is the result of applying uniform crossover to the strings  $\alpha_i$  and  $\beta_i[1, l_1]$ ; and  $q \in [l_1, l_2]$  is randomly selected. We consider  $\beta[l_1 + 1, l_1] = \varepsilon$ .

*Example 7.* Let  $\alpha_i = 011$  and  $\beta_i = 100011$  be two variable strings over the alphabet  $A = \{0, 1\}$ . In this case, we apply uniform crossover to  $\alpha_i$  and  $\beta_i[1, 3] = 100$ . Let us suppose that 111 is the resulting substring. This substring is the first part of the child. Now if  $q$  is, for instance, 4, the second part of the child is the string 0; and the complete child is  $\gamma_i = 1110$ .

**mutation:** The mutation process applied to a chromosome  $\bar{\alpha} = (\alpha_1, \dots, \alpha_m)$ , consists in randomly selecting a natural number  $q \in [1, 3|\bar{\alpha}|]$  and then perform one of the following operations:

1. if  $q \leq |\bar{\alpha}|$  then flip bit  $\alpha_i[j]$  where the pair  $(i, j)$  satisfies  $q = j + \sum_{k=1}^{i-1} |\alpha_k|$
2. if  $|\bar{\alpha}| < q \leq 2|\bar{\alpha}|$  then insert a randomly chosen symbol in alphabet  $A$  just before symbol  $\alpha_i[j]$ , where the pair  $(i, j)$  satisfies  $q = |\bar{\alpha}| + j + \sum_{k=1}^{i-1} |\alpha_k|$
3. otherwise, delete symbol  $\alpha_i[j]$  where the pair  $(i, j)$  satisfies  $q = 2|\bar{\alpha}| + j + \sum_{k=1}^{i-1} |\alpha_k|$

This mutation is applied to a chromosome  $\bar{\alpha}$  with a probability  $p$  which is a parameter of the evolutionary algorithm. Note that when mutation is applied to  $\bar{\alpha}$  the result is a chromosome  $\bar{\beta}$  such that  $LD(\bar{\alpha}, \bar{\beta}) = 1$ .

### 3.4 Local Search Procedure

We describe below a local search procedure suggested by Levenstein distance and inspired in common local search strategies for 0-1 binary strings. When Levenstein distance is used, for each gene of a chromosome there is more than one chromosome in its neighborhood of ratio 1. Note that this does not happen when Hamming distance is used and the alphabet is  $A = \{0, 1\}$ . As a consequence, an exhaustive local search procedure directed by Levenstein metric, will explore a larger portion of the search space. A non exhaustive variant of this process can be constructed by considering for each gene just one of the chromosomes in the neighborhood, randomly selected.

In order to formalize the above described process we proceed as follows. For any symbol  $b$  in the alphabet  $A = \{0, 1\}$ , let us denote by  $\bar{b}$  the result of flipping  $b$ . Let  $S = \{E_1, \dots, E_n\}$  be a word equation system over alphabet  $A = \{0, 1\}$  with set of variables  $\Omega = \{x_1, \dots, x_m\}$  and let  $\bar{\alpha} = (\alpha_1, \dots, \alpha_m)$ , be a chromosome. For each pair  $(i, j)$ ,  $1 \leq i \leq m$ ;  $1 \leq j \leq |\alpha_i|$ , denote by  $U(\alpha_i, j)$  the set consisting in the following binary strings:

1.  $\beta_i^1 = \alpha_i[1, j - 1]\bar{\alpha}_i[j]\alpha_i[j + 1, |\alpha_i|]$  (flip the bit  $\alpha_i[j]$ ).
2.  $\beta_i^2 = \alpha_i[1, j - 1]0\alpha_i[j, |\alpha_i|]$  (insertion of symbol 0 just before  $\alpha_i[j]$ ).
3.  $\beta_i^3 = \alpha_i[1, j - 1]1\alpha_i[j, |\alpha_i|]$  (insertion of symbol 1 just before  $\alpha_i[j]$ ).
4.  $\beta_i^4 = \alpha_i[1, j - 1]\alpha_i[j + 1, |\alpha_i|]$  (deletion of symbol  $\alpha_i[j]$ ).

At each iteration step the *exhaustive* local search procedure goes through the positions of the actual chromosome  $\bar{\gamma}$  and for each position  $(i, j) \in \{1, \dots, m\} \times \{1, \dots, |\gamma_i|\}$ , replaces  $\gamma_i$  by the element in  $U(\gamma_i, j)$  that produces the chromosome with best fitness, if and only if this element improves the fitness of the current chromosome  $\bar{\gamma}$ . On the other hand, the *non exhaustive* local search procedure randomly picks some element in  $U(\gamma_i, j)$  which replaces  $\gamma_i$  if there is an improvement in the fitness. This is done by an auxiliary procedure called *modify\_gen*. In both cases, for each  $i$ , once the case  $j = |\gamma_i|$  has been treated, we try to obtain some gain by adding a symbol as suffix of  $\gamma_i$ . This process iterates until there is no gain. Below we display the pseudo-code of these local search procedures. The differences between the exhaustive and non exhaustive case depends of the definition of the auxiliary procedure *modify\_gen* (exploring all possibilities or just one randomly selected, as indicated above).

```

Procedure Local_search (cr=(cr_1, ..., cr_m))
begin
repeat

```

```

crAux:=cr;
for i=1 to m do
  j:=0;
  while j<>|cr_i| do
    j:=j+1;
    cr_i:= modify_gen(i, j, cr);
  cr_i:=add_suffix(i, cr)
until cr = crAux
end

```

## 4 Experimental Results

We have executed our local search genetic algorithms, *LSG1*, which stands for the exhaustive version of local search, and *LSG2* (non exhaustive local search) on two sets of WES instances. The first set of instances is included in the library proposed in [1] and [2]. These problem instances, denoted as  $pn-m-q$ , consist of word equation systems with  $n$  equations,  $m$  variables and having a solution of length  $q$ . We run our program for various upper bounds of variable length  $d \geq q$ . Let us note that,  $m$  variables and  $d$  as upper bound for the length of a solution, determines a search space of size

$$\left(\sum_{i=0}^d 2^i\right)^m = (2^{d+1} - 1)^m \quad (4)$$

The second set is a set of 8 instances randomly generated forcing solvability. These are large instances, up to  $n = 50$ ,  $m = 35$ ,  $q = 24$ .

For the first set of problem instances we have executed our algorithm including as parameter an upper bound for the length of the solution in order to reproduce the conditions of the experiments shown in [1] and [2]. However, for the second set of problem instances, we start from an initial population in which the length of any chromosome has a small universal constant value as upper bound. The exact value of this constant is meaningless. The idea is that we start from an initial set of empty words and want to reach a solution evolving from this simple situation.

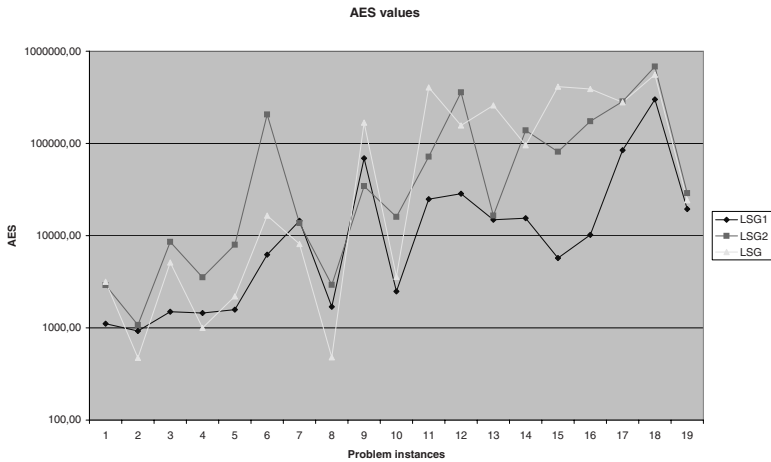
After some previous experimentation we have set the parameters of *LSG1* and *LSG2* to the following values: population size equals 5 and probability of mutation equals 0.75. It may seem something singular such a small population size and such a large probability of mutation but, analyzing many of the evolutionary algorithms for solving 3-SAT problem, ([12], [5] or [7]), we see that all of them have the property of very small size of population (frequently size one) and very large probability of mutation (sometimes 1). This fact is also pointed out in [1].

In all the executions, the algorithm stops if a solution is found or the limit of 1500000 evaluations is reached. The results of our experiments are displayed in Tables 1 and 2, based on 50 independent runs for each instance. The performance of the algorithm is measured first of all by the *Success Rate (SR)*, which represents the portion of runs

<sup>1</sup> Public available on line in <http://www.aic.uniovi.es/Tc/spanish/repository.htm#testproblems>.

**Table 1.** Experimental results for the first set of problem instances with various sizes of search space (S.S.). We have run for the instances *LSG1* (SR1 & AES1) and *LSG2* (SR2 & AES2). We also include the best results obtained in [2] (SR & AES). The elements of column U.B. are the different upper bounds.

P. instance	U.B.	S.S.	SR1	SR2	SR	AES1	AES2	AES
p10-8-3	3	2 <sup>32</sup>	100%	100%	100%	1108	2905.2	3164.24
p25-8-3	3	2 <sup>32</sup>	100%	100%	100%	927.6	1077.4	473.46
p10-8-3	4	2 <sup>40</sup>	100%	100%	100%	1500.55	8574.7	5121.25
p25-8-3	5	2 <sup>48</sup>	100%	100%	100%	1445.4	3533	1002.26
p5-8-3	6	2 <sup>56</sup>	100%	100%	100%	1580	7950.4	2193
p5-15-3	3	2 <sup>60</sup>	100%	100%	100%	6235	206387	16459.25
p10-15-3	3	2 <sup>60</sup>	100%	100%	100%	14513.33	13685.22	8124.48
p15-12-4	4	2 <sup>60</sup>	100%	100%	100%	1690	2932.9	479.63
p10-8-3	7	2 <sup>64</sup>	100%	100%	100%	69046.62	34371.8	168344
p25-8-3	8	2 <sup>72</sup>	100%	100%	100%	2480.3	15924.6	3567.86
p10-8-3	10	2 <sup>88</sup>	100%	100%	85%	24892.7	71918.55	405326
p5-15-3	5	2 <sup>90</sup>	100%	90%	100%	28432.9	357109.77	156365.52
p10-15-3	5	2 <sup>90</sup>	100%	100%	100%	14851.88	16492.7	258556
p10-15-5	5	2 <sup>90</sup>	100%	100%	100%	15498.8	138438.2	95457.93
p25-23-4	4	2 <sup>115</sup>	100%	100%	100%	5720.43	81389.33	412375
p25-23-4	5	2 <sup>138</sup>	100%	96%	81%	10210.25	174318.37	389630
p15-25-5	5	2 <sup>150</sup>	90%	86%	98%	84708.33	284796.25	278782.36
p5-15-3	10	2 <sup>165</sup>	80%	30%	8%	299770.87	681720.66	557410.58
p25-8-3	20	2 <sup>168</sup>	100%	100%	100%	19377.4	28939.6	24221

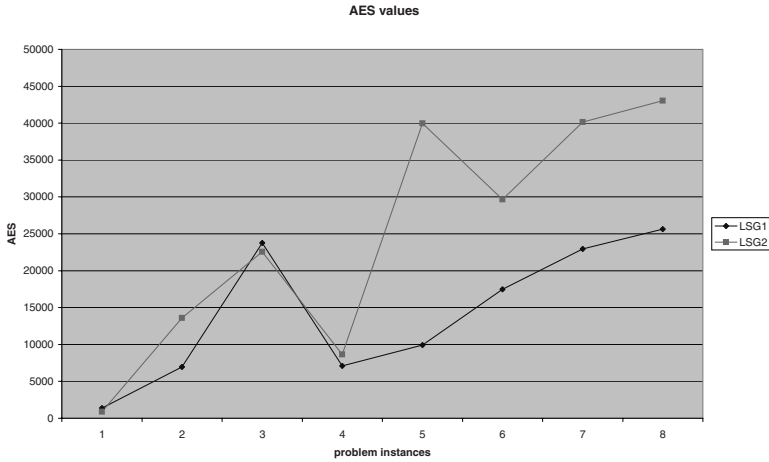


**Fig. 1.** AES values corresponding to table 1

where a solution has been encountered. As a measure of time complexity, we use the *Average number of Evaluations to Solution (AES)* index, which counts the average number of fitness evaluations performed up to find a solution in successful runs.

**Table 2.** Experimental results for the second set of problem instances

P. instance	SR1	SR2	AES1	AES2
p15-5-8	100%	100%	1393.2	880.6
p50-25-10	100%	100%	6961.6	13594.4
p15-7-13	100%	100%	23772.77	22562.2
p20-12-18	100%	100%	7109.6	8646.2
p40-20-19	100%	100%	9933.1	39981.88
p60-16-24	100%	100%	17463.5	29676.9
p50-30-20	100%	100%	22944.8	40151.7
p50-35-20	100%	100%	25623	43027.55



**Fig. 2.** AES values corresponding to table 2

For the first set of problem instances we compare our two local search procedures, *LSG1* and *LSG2* with the procedure *LSG* described in [2], which is based on Hamming distance. We observe that *LSG1* diminishes the number of evaluations of *LSG2* and *LSG*. This can be confirmed by looking at the AES values reported in table 1 and figure 1.

For the second set of problem instances, containing larger problems, we remark that algorithm *LSG* has a very small SR after more than 1500000 evaluations. However both algorithms *LSG1* and *LSG2* obtain a very good SR within a very small AES, as it is shown in table 2 and fig. 2. It is also remarkable the following fact: instances reported in table 2 are larger instances than those reported in table 1; however, in many cases, the AES values obtained for this second set of problems are considerably better than the corresponding AES values obtained for the first set of problem instances. We conclude that, experimentally, it seems to be more efficient to start from small populations with short chromosomes than using small populations with large chromosomes.

A careful analysis of figure 1 and figure 2 does not allow us to infer a clear relation between size of search space and AES values. We conjecture that our local search



procedures quickly find a solution, even if the size of the search space is high, for those non ill conditioned instances having short solutions.

## 5 Summary and Conclusions

In this paper we have presented a genetic algorithm for solving word equation systems which incorporates a local search procedure based on Levenstein distance. On a large set of problems, we have shown that our algorithm is capable of obtaining high quality solutions for large problems of various characteristics. The use of a local search procedure improves the quality of the individuals and directs the GA to the some solution if it exists. Moreover, experimental results show that Levenstein distance is better suited, to direct the local search, than Hamming distance as used in [2].

We want to remark that our method is not compared with any exact deterministic algorithm because due to the very high complexity of WES problem no implementation of deterministic algorithms is available solving large instances. A very recent paper by Plandowski (see [14]) provides a deterministic EXPTIME algorithm for WES. Plandowski's algorithm can be considered "theoretically" efficient in the sense that the lower complexity exponential bounds for the WES problem are reached by this algorithm. It would be of practical interest to know some information about the probability distribution of solutions of randomly generated WES instances, in the same spirit of results concerning hard 3-SAT instances. Such kind of results could direct experimentation to a better understanding of the time complexity and the success rate of our algorithm.

## References

1. Alonso, C.L., Drubi, F., Montaña, J.L.: An Evolutionary Algorithm for Solving Word Equation Systems. In: Conejo, R., Urretavizcaya, M., Pérez-de-la-Cruz, J.L. (eds.) *Current Topics in Artificial Intelligence*. LNCS (LNAI), vol. 3040, pp. 147–156. Springer, Heidelberg (2004)
2. Alonso, C.L., Drubi, F., Gómez-García, J., Montaña, J.L.: Word Equation Systems: The Heuristic Approach. In: Bazzan, A.L.C., Labidi, S. (eds.) *SBIA 2004*. LNCS (LNAI), vol. 3171, pp. 83–92. Springer, Heidelberg (2004)
3. Angluin, D.: Finding patterns common to a set of strings, *J. J. C. S. S.* 21(1), 46–62 (1980)
4. Baader, F., Siekmann, J.H.: Unification Theory. In: *Handbook of Logic in Artif. Int. and Logic Prog.*, vol. 2, Clarendon Press, Oxford (1994)
5. Eiben, A., van der Hauw, J.: Solving 3-SAT with adaptive Genetic Algorithms. In: *4th IEEE Conference on Evolutionary Computation*, pp. 81–86. IEEE Press, Los Alamitos (1997)
6. Goldberg, D.E.: *Genetic Algorithms in Search Optimization & Machine Learning*. Addison Wesley Longman, London (1989)
7. Gottlieb, J., Marchiori, E., Rossi, C.: Evolutionary Algorithms for the Satisfiability Problem. *Evolutionary Computation* 10(1) (2002)
8. Gutiérrez, C.: Satisfiability of word equations with constants is in exponential space. In: *Proc. FOCS 1998*, IEEE Computer Society Press, California (1998)
9. Gutiérrez, C.: Satisfiability of equations in free groups is in PSPACE. In: *STOC 2000. Theory of Computing*, pp. 21–27. ACM Press, New York (2000)
10. Levenshtein, V.I.: Bynary Coded Capable of Correcting Deletions, Insertions and Reversals. *Doklady Akademii Nauk SSR* 163(4), 845–848 (1965)

11. Makanin, G.S.: The Problem of Solvability of Equations in a Free Semigroup. *Math. USSR Sbornik* 32(2), 129–198 (1977)
12. Marchiori, E., Rossi, C.: A Flipping Genetic Algorithm for Hard 3-SAT Problems. In: Banzhaf, W., et al. (eds.) *Proceedings of G.E.C.C.O.*, pp. 393–400 (1999)
13. Plandowski, W.: Satisfiability of Word Equations with Constants is in PSPACE. In: *FOCS 1999*, pp. 495–500 (1999)
14. Plandowski, W.: An efficient algorithm for solving word equations. In: *Annual ACM Symposium on Theory of Computing archive Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pp. 467–476. ACM Press, New York (2006)
15. Plandowski, W., Rytter, W.: Application of Lempel-Ziv encodings to the Solution of Words Equations. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) *ICALP 1998. LNCS*, vol. 1443, pp. 731–742. Springer, Heidelberg (1998)
16. Robson, J.M., Diekert, V.: On quadratic Word Equations. In: Meinel, C., Tison, S. (eds.) *STACS 99. LNCS*, vol. 1563, pp. 217–226. Springer, Heidelberg (1999)
17. Srivas, M.K., Miller, S.P.: formal Verification of the AAMP5 Microprocesor. In: Hinchey, M.G., Bowen, J.P. (eds.) *Application of Formal Methods, International Series in Computer Science*, ch.7, pp. 125–180. Prentice-Hall, Hemel Hempstead (1995)
18. Ukkonen, E.: On-line construction of suffix trees. *Algorithmica* 14(3), 249–260 (1995)

# Simple Model-Based Exploration and Exploitation of Markov Decision Processes Using the Elimination Algorithm

Elizabeth Novoa

Departamento de Ingeniería Informática, Universidad de Santiago de Chile,  
Av. Ecuador 3659, Santiago, Chile  
elizabeth.novoa@correo.usach.cl

**Abstract.** The fundamental problem in learning and planning of Markov Decision Processes is how the agent explores and exploits an uncertain environment. The classical solutions to the problem are basically heuristics that lack appropriate theoretical justifications. As a result, principled solutions based on Bayesian estimation, though intractable even in small cases, have been recently investigated. The common approach is to approximate Bayesian estimation with sophisticated methods that cope the intractability of computing the Bayesian posterior. However, we notice that the complexity of these approximations still prevents their use as the long-term reward gain improvement seems to be diminished by the difficulties of implementation. In this work, we propose a deliberately simplistic model-based algorithm to show the benefits of Bayesian estimation when compared to classical model-free solutions. In particular, our agent combines several Markov Chains from its belief state and uses the matrix-based Elimination Algorithm to find the best action to take. We test our agent over the three standard problems Chain, Loop, and Maze, and find that it outperforms the classical Q-Learning with e-Greedy, Boltzmann, and Interval Estimation action selection heuristics.

## 1 Introduction

In the realm of sequential decision making under uncertainty in general and reinforcement learning (RL) in particular, the fundamental problem an agent has to face is how to explore and exploit the environment [1]. Namely, an exploring action tries to find a better estimate of the environment dynamics by paying the cost of search, whereas there is a conflicting desire of just exploiting the current estimation. Classical solutions to the so-called exploration/exploitation tradeoff are basically heuristics fast and easy to understand. They have been widely applied in all kind of settings showing a decent performance given a fine tuning of their parameters [2]. However, they are difficult to quantify, do not perform well in all circumstances, and lack appropriate theoretical justifications. They seem to resemble early stages of currently, theoretically strong fields, such as machine learning, in that they are nothing but highly sophisticated intuitions with a strong dependency on manual tuning [3,4,5,6].

The main source of difficulties in the exploration/exploitation tradeoff comes from not keeping an account of the uncertainty. We can see this by examining the (classical) Q-Learning algorithm [7] which is still heavily used because of its simplicity and relatively good performance [8]. Q-Learning does not distinguish between action selection and planning. In planning, the agent should give bonuses to courses of actions with high uncertainty more than assigning myopic values based on the available information. This is assessed at some level by action selection. For example, the  $\epsilon$ -Greedy heuristic performs a random action with probability  $\epsilon$ , the Boltzmann heuristic samples a random action according to an exponential distribution based on the Q values, and Interval Estimation [9,10] assigns a bonus based on the confidence on Q values. However, a more explicit account of the uncertainty is needed by means of a model of the environment.

There is a Bayesian solution to learn and plan optimally provided an appropriate model of the environment [11]. The solution is conceptually correct and solves implicitly, and correctly the tradeoff between exploration and exploitation. The definitive drawback is that Bayesian reinforcement learning is computationally intractable, even in trivial cases [12,13]. Nevertheless, there have been recent substantial amount of research on this topic [14,15,4,16,17,18,19,20]. Most of the approaches use the Markov Decision Process (MDP) framework to represent the dynamics of the environment. Thus, the Bayesian agent keeps a belief over these dynamics as a meta-level MDP defined by probability distributions over transitions and rewards. To cope the notorious intractability, all the methods can be roughly summarized as different means to approximate the posterior of the Bayesian update.

Despite the correctness and reported good performance of these approximate Bayesian solutions, there persist a belief that they are not significant if one considers the complexity of implement them [4]. Indeed, this is not too far from reality. Some of the approximations need access to linear programming software packages [14], or solve complete sets of Bellman equations by slow iteration algorithms [18]. Some near-optimal approaches need to use an exponential amount of memory and time if one wants a good solution [16], or are still too theoretical to be implemented even on standard toy problems [15,17]. Therefore, the aim of this work is to offer a principled solution that can be closer to the implementation simplicity of Q-Learning, but that performs a much better exploration and exploitation. We use the elegant and simple recent Sonin's Elimination Algorithm [21] to select an action from a combination of Markov Chains. Through experiments over standard problems, we show that our algorithm outperforms the classical Q-Learning with  $\epsilon$ -Greedy, Boltzmann, and Interval Estimation action selection heuristics.

## 2 Background

Reinforcement Learning (RL) is learning through the direct interaction with an uncertain environment. Since it does not assume the existence of a teacher that gives examples to learn, it is a form of trial-and-error learning: the agent repeatedly chooses actions over states of the environment, and receives rewards.

The RL aims to learn a policy  $\pi$  commanding the action to take at each state such that its total discounted reward is maximized. Formally, if a reward  $r(x_t)$  is awarded when the environment is in state  $x_t \in X$  at time  $t$  and  $\gamma$  is the discount factor, the total discounted reward is  $r(x_t) + \gamma r(x_{t+1}) + \gamma^2 r(x_{t+2}) + \dots$ . The crucial point of RL is that the agent must estimate the conditions that lead to reinforcements or punishments only using the state-reward pairs of experiences.

## 2.1 Classics Solutions to the Reinforcement Learning Problem

Since Watkins introduced it in 1989, Q-Learning [7] is of the most widely used methods to solve reinforcement learning problems. In Q-Learning, an estimate of the action-value

$$\hat{Q}(x, a) \leftarrow (1 - \alpha)\hat{Q}(x, a) + \alpha(r + \gamma \max_{a'} \hat{Q}(x', a')) \quad (1)$$

is updated each time the agent performs action  $a$  in state  $x$ , perceives a reward  $r$ , and reaches state  $x'$ . The *learning rate*  $\alpha$  balances the uncertainties from state transitions, rewards, and errors in estimates of  $Q$ .

At each decision time, the agent tries to combine exploration of the environment and exploitation of the estimates  $Q$  by selecting an action. The most common action selecting heuristics are  $\epsilon$ -Greedy, Boltzmann, and Interval Estimation [4]. Q-Learning is very effective, but requires a large number of trials to estimate the true  $Q$  values.

**$\epsilon$ -Greedy.** Selects the action  $a^* = \arg \max_a \hat{Q}(x, a)$  with probability  $1 - \epsilon$ . Otherwise, select random action  $a \in A$ .

**Boltzman.** Sample a random action from an exponential distribution such that

$$a \sim \frac{\exp \left\{ \tau^{-1} \hat{Q}(x, a) \right\}}{\sum_{a' \in A} \exp \left\{ \tau^{-1} \hat{Q}(x, a') \right\}},$$

where  $\tau$  is a *temperature* parameter.

**Interval Estimation.** We choose an action

$$a = \arg \max_a \left\{ \hat{Q}(x, a) + U(x, a) \right\},$$

where  $U(x, a)$  is a  $(1 - \alpha)$  upper confidence interval of  $\hat{Q}(x, a)$  [9,10].

## 2.2 Bayesian Reinforcement Learning

The Bayesian Reinforcement Learning in turn keeps a belief about the reward and transition probability distribution of the underlying Markov Decision Process (MDP). This is sometimes called a Meta-MDP [18] as each sample from the reward and transition distributions is a MDP.

Formally, the agent keeps a distribution over the possible transitions  $T$  based on a belief  $\mathbf{b}$  such that

$$T(x, a, x') = P(x'|x, a, \mathbf{b}), \quad (2)$$

where  $x'$  is the state reached after taking action  $a$  in state  $x$ , and  $\mathbf{b}$  is the belief about this distribution. Similarly, the agent keeps a distribution over possible rewards

$$R(x, a) \sim P(r|x, a, \mathbf{b}), \quad (3)$$

where  $R$  is a scalar value representing the amount of reward received after taking action  $a$  in state  $x$ . This reward is sampled from the distribution of possible rewards based on the belief  $\mathbf{b}$ .

For the sake of computational complexity, there is a first-order Markov assumption in transitions, i.e. they are affected by the immediate previous action and state, and the transition and reward distributions are independent, i.e. the transition probability does not depend on the reward probability given the action taken. Therefore, we can express our joint belief of  $R$  and  $T$  as

$$P(R, T|\mathbf{b}) = \prod_x \prod_a P(x'|x, a, \mathbf{b})P(r|x, a, \mathbf{b}).$$

### 3 Method

#### 3.1 The Optimal Stopping in a Markov Chain and the Elimination Algorithm

We use the following notation. A Markov chain (MC) is defined by  $M = (X, P)$ , where  $X$  is a countable state space, and  $P = p(x, y)$  is a transition matrix between states. The tuple  $M = (X, P, r(x), \gamma(x))$ , where  $r(x)$  is a one step reward function, and  $\gamma(x)$  is a discount factor,  $0 < \gamma(x) < 1$  and  $x \in X$ , is called a *reward model with termination*. We assume that the state space  $X$  contains an absorbing point  $x_*$  ( $p(x_*, x_*) = 1$ ), and therefore that the function  $\gamma(x)$  ( $\forall x \in X$ ) can be seen as the probability of *survival* at state  $x$  such that  $1 - \gamma(x) = p(x, x_*)$ . Using this absorbing state, the probability transition matrix  $P$  has embedded the function  $\gamma(x)$ . We assume  $r(x_*) = 0$ . Let  $v(x)$  be the value function for the reward model defined as

$$v(x) = \sup_{\tau > 0} \frac{E_x \left[ \sum_{t=0}^{\tau-1} r(x_t) \prod_{j=0}^t \gamma(x_j) \mid x_0 = x \right]}{E_x \left[ \sum_{t=0}^{\tau-1} \prod_{j=0}^t \gamma(x_j) \mid x_0 = x \right]}. \quad (4)$$

The value of  $\tau$  for which the supremum is found in Equation 4 is the optimal stopping time if we start from state  $x$ . This means that it is better to stop the Markov Chain *before* the time  $\tau$  is reached, otherwise the increasing proportion between the total discounted reward (the denominator in Equation 4) and the accumulated probability of surviving (the numerator in Equation 4) decays from

time  $\tau$  and thereafter. A different interpretation would be that the risk of receiving a one step reward at time  $\tau$  exceeds the benefits of that reward, and therefore it is better to stop after time  $\tau - 1$ .

Sonin proposes a recursive algorithm [21] based on Gaussian elimination in matrix  $P$  to compute  $v$ . The intuition behind it is as novel as simple: do not care where to stop, but where *not* to stop. The algorithm recursively *eliminates* states where we should not stop and incorporates the rewards and transitions of the eliminated states into the remaining states'. The last set of remaining states (which have the same value of  $v$ ) is the optimal stopping set. Another important aspect of the algorithm is that it only uses matrix computations to find the values  $v(x)$ .

A result of the Elimination Algorithm we use in the Markov Decision Process context —an actually, the main motivation in [21]— it that, if we need to choose one among several (independent) reward models to gather reward, it is optimal to choose the corresponding Markov Chain with highest value  $v$ .

### 3.2 The Proposed Algorithm

Our algorithm works by sampling a maximum likelihood set of  $K$  Markov Chains  $\{M_i\}_{1 \leq i \leq K}$  from the current belief  $\mathbf{b}$  with fixed, uniform random policies  $\pi_1, \dots, \pi_K$ . The agent then computes the  $v$  values and uses the policy

$$\pi^*(x) = \arg \max_{\pi_i} \{v_i(x, \pi_i) : i = 1, \dots, K\} \quad (5)$$

to perform the action  $\pi^*(x)$  in current state  $x$ . We relax the computation of  $v$  by creating a horizon  $\tau \leq H$  in every Markov Chain and considering only  $C$  possible next states that can be reached from any state. Every time the  $C$  connections are made, there is a set of states not considered in the expansion whose transitions and rewards are incorporated into the *parent* state using the ideas of the Elimination Algorithm.

With a little work, we can see how we construct a reward model (Markov Chain) using policy  $\pi$ . First, we create the generating state  $x_t$  ( $x_0 = x$ ) and attach to it our current belief  $\mathbf{b}$  and a reward  $r(x_0) = 0$ . Then, for each possible next-state  $x'$  in  $T(x'|x_t, \pi(x_t), \mathbf{b})$ , we create a new state  $x_{t+1}$ , attach to it the new belief  $\mathbf{b}_{t+1}$ , and reward  $r(x_{t+1}) = E[R(r|x_t, \pi(x_t), \mathbf{b})]$ ; and create a stochastic transition  $p(x_{t+1}, x_t)$  between  $x_t$  and  $x_{t+1}$  equal to  $\gamma T(x_{t+1}|x_t, \pi(x_t), \mathbf{b})$ . Additionally, every state  $x$  is connected to the absorbing state  $x_*$  with  $p(x_*, x) = 1 - \gamma$ . The belief  $\mathbf{b}_{t+1}$  means the posterior belief if we had moved from state  $x_t$  to  $x_{t+1}$ . Then, we draw  $C$  states  $x_{t+1}$  from  $T(\cdot|x_t, \pi(x_t), \mathbf{b})$ , eliminate the remaining states by modifying  $p(x_{t+1}, x_t)$  and  $r(x_t)$  according to the Elimination Algorithm. For every of the new  $C$  states  $x_{t+1}$ , we keep generating the MC using the same procedure. When the horizon  $H$  is reached, instead of attaching a reward  $E[R(r|x_t, \pi(x_t), \mathbf{b})]$  to  $x_{t+1}$ , we attach the mean of the highest previous computations of  $v$  on state  $x_{t+1}$ , denoted by  $U(x_{t+1})$ . This serves us to optimistically *chain* results from previous Markov Chains. For the avid reader, this value  $U(x_{t+1})$  is an attempt to approximate the value of a state, given that the  $v(x, \pi)$

is a lower bound of the action-value  $Q(x, \pi(x))$  in the classical Bellman formulation of the reinforcement learning problem. We call  $U(\cdot)$  the pseudo-values of states. Formally, our relaxation in horizon and value of states can be seen in (4) as

$$v(x, \pi) = \sup_{0 < \tau \leq H} \frac{E_{\pi} \left[ \sum_{t=0}^{\tau-1} r(x_t) \prod_{j=0}^t \gamma(x_j) \mid x_0 = x \right]}{E_{\pi} \left[ \sum_{t=0}^{\tau-1} \prod_{j=0}^t \gamma(x_j) \mid x_0 = x \right]}, \tag{6}$$

where  $r(x_{H-1}) = U(x_{H-1})$ .

Notice that all this planning happens into the agent’s mind and only after the action is actually performed the values of the approximate Bayesian inference process are updated. This is, the reward, next state, and the highest  $v$  is used to update the belief on rewards, transitions, and  $U(x)$ , respectively. Figure 1 shows the proposed algorithm.

---

**BAYESIAN ELIMINATION ALGORITHM** ( $K$  = number of arms,  $C$  = number of next-states to sample (from current belief) at each state,  $H$  = horizon,  $x_0$  = starting state,  $\mathbf{b}$  = belief about transitions and rewards)

- Pseudo-value of states  $U(\cdot) \leftarrow \mathbf{0}$
  - $x \leftarrow x_0$
  - While (not stop criterion)
    1. Repeat  $K$  times
      - Create Markov Chain  $i$  using a Bayesian agent that sample one action  $a$  uniformly from the current state  $x$  and connect it to  $C$  possible  $x'$  next states sampled from the current belief  $T(x'|x, a, \mathbf{b})$ , and attach reward  $r(x') = E[R(x'|x, a, \mathbf{b})]$ . Repeat recursively until horizon  $H$  is reached. Attach reward  $U(x_{H-1})$  to states in the horizon.
    2. Compute the  $\{v_i(x, \pi_i)\}_{1 \leq i \leq K}$  values of Markov Chains according to (6) using the Elimination Algorithm
    3. Perform action instructed by policy of Markov Chain with highest  $v$  value according to (5).
    4. Update belief for transition and reward based on  $x, a, x'$ , and reward  $r$ . If this is the  $(n + 1)$ -th visit to  $x$ , update the pseudo-value of state using  $U(x) \leftarrow (n + 1)^{-1}(nU(x) + \max_i v_i)$ .
- 

**Fig. 1.** The algorithm proposed. See the description in the text for details.

## 4 Experiments

### 4.1 Choosing the Belief Model of Our Algorithm

For computational simplicity, we have chosen a conjugate prior of our conditional such that the posterior belong to the same conjugate family. We have chosen a mixture of Dirichlet distributions as the belief and conditionals of transitions. This is, for each action-state pair, we have  $\mathbf{b}_{x,a}^{T^-} = Dir(\alpha_1, \dots, \alpha_{x'}, \dots, \alpha_n)$ ,



where  $\alpha_{x'}$  represents the number of times the transition between state  $x$  and  $x'$  has been seen. After looking at the evidence —the next state  $x'$  reached after taking the action—, the agent will update its belief as  $\mathbf{b}_{x,a}^{T+} = Dir(\alpha_1, \dots, \alpha_{x'} + 1, \dots, \alpha_n)$ . For all Dirichlet distributions, we assume a uniform distribution with all  $\alpha$ s equal to 1 at time zero.

Similarly for computational complexity, we have chosen a mixture of Gaussian distributions with unary variance as the belief and conditions of the rewards. For each action-state pair, we have that  $\mathbf{b}_{s,a}^{R-} = N(\bar{r}, 1)$ . After looking at the  $(n + 1)$ -th evidence reward  $r$ , the agent will update his belief using  $\mathbf{b}_{s,a}^{R+} = N((n + 1)^{-1}(n\bar{r} + r), 1)$ . For all Gaussians, we assume a prior with  $\bar{r} = R_{\max}$  (i.e. maximum reward available at any state-action).

### 4.2 Test Problems

The agent is tested in three standard problems from [18/20].

**Chain.** Figure 2 shows the chain problem, which consists of five states. There are two actions  $a$  and  $b$  available for the agent, but, with probability 0.2, the action will have resulted in an opposite effect. The optimal policy for this problem is to perform action  $a$  at every state and would generate a total reward of 3677 after 1000 steps. However, the agent can be misled by the immediate rewards from taking action  $b$ . Therefore, the agent needs to explore effectively and estimate the discounted reward accurately.

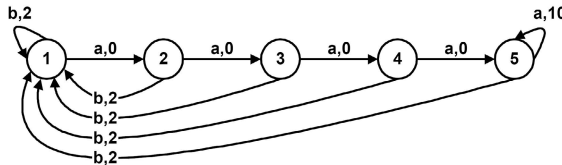


Fig. 2. The Chain Problem

**Loop.** This problem consists of two 5-state loops, which are connected at a single start state, as shown in Figure 3. Two deterministic actions are available. Since taking action  $b$  at every state in the left loop yields a reward of 2, the optimal policy is to perform action  $b$  everywhere and would yield a total reward of 400 after 1000 steps. To attain the optimal policy, the agent needs to find the middle ground between exploration and exploitation, but it is difficult because the agent can get trapped in the right loop to obtain a series of smaller rewards.

**Maze.** This problem (Figure 4) represents the learning and planning in a potentially big state space. The agent can choose among four actions (left, right, up, or down) and advance one square in the maze. If it hits a wall, the action does not have effect. The problem is to move from the top-left (start) corner to the top-right (goal) corner collecting the three flags on the way.

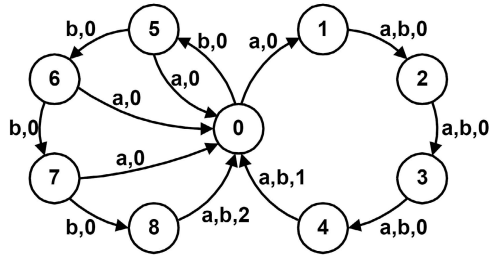


Fig. 3. The Loop Problem

The only time the agent receives reward is at the goal, when a unary reward will be awarded for each flag collected. Once reached the goal, the agent is immediately returned to the start. An additional difficulty is that, with probability 0.1, the actual action of the agent might take a perpendicular direction of the intended action. To reduce the complexity of the algorithms, the agent will have limited layout information by identifying only the immediate successors of each state. For estimated optimal policy would yield an accumulated reward of 1860 each 20000 steps.

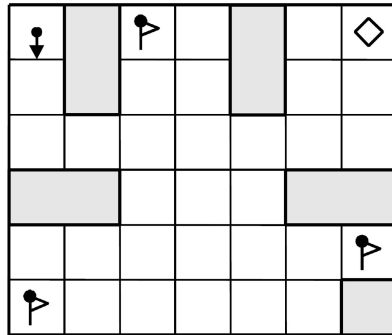


Fig. 4. The Maze Problem

### 4.3 Results

The standard setting to test these problems is as follows. The experimental results are shown as accumulated totals of reward received over learning phases of 1000 steps for Chain and Loop, and 20000 steps for Maze. The averages were taken over 256 runs for Chain and Loop, and 16 runs for Maze. Table 1 summarizes the performance in phases 1, 2, and 8. The parameters for our algorithm (shown as Bayesian EA in Table 1) are  $H = 3$ ,  $C = 3$ , and  $K = 50$  for Chain and Loop, and  $H = 4$ ,  $C = 3$ ,  $K = 100$  for Maze. The results show that the performance of our algorithm is significantly better than the primitive Q-Learning with the action-selection strategies tested. Although Q-Learning+ $\epsilon$ -Greedy does

**Table 1.** Performance of models during different phases. For problems Chain, and Loop each phase comprises 1000 steps. For the Maze problem, each phase comprises 20000 steps. Best performance is shown in bold-typeface. Parts of the results for Boltzmann and Interval Estimation (IE) comes from [20]. The algorithm proposed is Bayesian EA (Elimination Algorithm).

CHAIN	PHASE 1	PHASE 2	PHASE 8
QL $\epsilon$ -GREEDY	2300 $\pm$ 400	2400 $\pm$ 100	2470 $\pm$ 150
QL BOLTZMANN	1606 $\pm$ 26	1623 $\pm$ 22	1890 $\pm$ 10
QL IE	2344 $\pm$ 78	2557 $\pm$ 90	2600 $\pm$ 7
<b>Bayesian EA</b>	<b>2900 <math>\pm</math> 10</b>	<b>3400 <math>\pm</math> 20</b>	<b>3570 <math>\pm</math> 23</b>
LOOP	PHASE 1	PHASE 2	PHASE 8
QL $\epsilon$ -GREEDY	198 $\pm$ 100	220 $\pm$ 92	226 $\pm$ 120
QL BOLTZMANN	186 $\pm$ 1	200 $\pm$ 1	210 $\pm$ 2
QL IE	264 $\pm$ 1	293 $\pm$ 1	310 $\pm$ 2
<b>Bayesian EA</b>	<b>395 <math>\pm</math> 5</b>	<b>399 <math>\pm</math> 1</b>	<b>399 <math>\pm</math> 1</b>
MAZE	PHASE 1	PHASE 2	PHASE 8
QL $\epsilon$ -GREEDY	<b>600 <math>\pm</math> 150</b>	1300 $\pm$ 100	1320 $\pm$ 150
QL BOLTZMANN	195 $\pm$ 20	1024 $\pm$ 29	1400 $\pm$ 10
QL IE	269 $\pm$ 1	253 $\pm$ 3	640 $\pm$ 2
<b>Bayesian EA</b>	230 $\pm$ 10	<b>1350 <math>\pm</math> 20</b>	<b>1450 <math>\pm</math> 30</b>

a quite good job in the first phase of Maze, our algorithm performs a better learning and yields better accumulated rewards in Phase 2 and thereafter.

## 5 Conclusions

In this work, we show a simple model-based approach to learn the dynamics of a MDP within a reinforcement learning framework. We tried to assess the fact that model-free solutions are still heavily used partially due to the complexity of implementing most of the model-based algorithms. Our solution uses the Sonin’s Elimination Algorithm to find the optimal stopping of a Markov Chain. This algorithm is very simple in that it needs only standard and widely available basic operations with matrices, which contrasts with the relative complexity of current model-based approaches. We compare our approach with the classical Q-Learning with different action selection heuristics over different standard problems. Our algorithm performs better than Q-Learning compelling to reconsider simple model-based approaches as a first step to correctly and elegantly solve the exploration/exploitation tradeoff.

## References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
2. Kaelbling, L.P., Littman, M.L., Moore, A.P.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 237–285 (1996)

3. Poupart, P., Vlassis, N., Hoey, J., Regan, K.: An analytic solution to discrete bayesian reinforcement learning. In: 23rd International Conference on Machine Learning (2006)
4. Wang, T., Lizotte, D., Bowling, M., Schuurmans, D.: Bayesian sparse sampling for on-line reward optimization. In: International Conference on Machine Learning, Bonn, Germany (2005)
5. Jordan, M.I. (ed.): Learning in graphical models. MIT Press, Cambridge (1999)
6. Neal, R.M.: Bayesian Learning for Neural Networks. Springer, New York (1996)
7. Watkins, C.: Learning from Delayed Rewards. PhD thesis, University of Cambridge (1989)
8. Russell, S.J., Norvig, P.: Artificial intelligence: a modern approach, 2nd edn. Prentice Hall/Pearson Education, Upper Saddle River, N.J (2003)
9. Wiering, M.: Explorations in efficient reinforcement learning. PhD thesis, University of Amsterdam (1999)
10. Kaelbling, L.P.: Associative reinforcement learning: Functions in k-dnf. Machine Learning 15(3), 279–298 (1994)
11. Duff, M.O.: Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes. PhD thesis, University of Massachusetts Amherst (2002)
12. Lusena, C., Goldsmith, J., Mundhenk, M.: Nonapproximability results for partially observable markov decision processes. JAIR 14, 83–103 (2001)
13. Mundhenk, M., Goldsmith, J., Lusena, C., Allender, E.: Complexity of finite-horizon markov decision process problems. Journal of the ACM (JACM) 47(4), 681–720 (2000)
14. Castro, P.S., Precup, D.: Using linear programming for bayesian exploration in markov decision processes. IJCAI , 2437–2442 (2007)
15. Brafman, R.I., Tennenholtz, M.: R-max - a general polynomial time algorithm for near-optimal reinforcement learning. J. Mach. Learn. Res. 3, 213–231 (2003)
16. Kearns, M., Mansour, Y., Ng, A.Y.: A sparse sampling algorithm for near-optimal planning in large markov decision processes. Machine Learning 49(2-3), 193–208 (2002)
17. Kearns, M., Singh, S.: Near-optimal reinforcement learning in polynomial time. Machine Learning 49(2-3), 209–232 (2002)
18. Strens, M.J.A.: A bayesian framework for reinforcement learning. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 943–950. Morgan Kaufmann, San Francisco (2000)
19. Dearden, R., Friedman, N., Andre, D.: Model based bayesian exploration. In: Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence, pp. 150–159 (1999)
20. Dearden, R., Friedman, N., Russell, S.J.: Bayesian q-learning. In: AAAI/IAAI, pp. 761–768 (1998)
21. Sonin, I.M.: A generalized gittins index for markov chain and its recursive calculation. submitted to Statistics and Probability Letters (2007)

# A Simple Model for Assessing Output Uncertainty in Stochastic Simulation Systems

Tengda Sun<sup>1,2,3</sup> and Jinfeng Wang<sup>1</sup>

<sup>1</sup> Institute of Geographic Science & Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, P.R. China

<sup>2</sup> Graduate University of the Chinese Academy of Sciences, Beijing 100039, P.R. China

<sup>3</sup> Navigation College, Jimei University, XIAMEN, 361021, P.R. China  
suntdd@lreis.ac.cn, wangjff@lreis.ac.cn

**Abstract.** The need for expressing uncertainty in stochastic simulation systems is widely recognized. However, the emphasis in uncertainty has been directed toward assessing simulation model input parameter uncertainty, while the analysis of simulation output uncertainty is deduced from the input uncertainty. Most recently used methods to assess uncertainty include Delta-Method approaches, Resampling method, Bayesian Analysis method and so on. The problem for all these methods is that the typical simulation user is not particularly proficient in statistics, and so is unlikely to be aware of appropriate sensitivity and/or uncertainty analyses. This suggests the need for a transparent, implementable and efficient method for understanding uncertainty, especially for simulation output uncertainty. In this paper, we propose a simple and straightforward framework to assess stochastic simulation output uncertainty based on Bayesian Melding. We firstly assume the form of probability distribution function of simulation output. We also assume that the final output uncertainty is the weight sum of uncertainty for every simulation output and the weight of each simulation run is proportional to its probability. The advantage of these assumptions is that to describe the simulation output uncertainty in the form of probability distribution function after limited simulation runs, we need only to do two things (1) to estimate parameters in the simulation output probability distribution function and (2) to calculate weight for each simulation. Both of them are discussed in detail in this paper.

**Keywords:** Simulation Output, Uncertainty Assessment, Stochastic Simulation System, Bayesian Melding.

## 1 Introduction

Simulation is a valuable tool in assessing the survivability and vulnerability of complex systems to natural, abnormal, and hostile events. However, there still remains the need to assess the accuracy of simulations by comparing computational predictions with experimental test data through the process known as validation of computational simulations. Physical experimentation, however, is continually increasing in cost and time required to conduct the test [1]. Uncertainty as a source of

nondeterministic behavior derives from lack of knowledge of the system or the environment. This restrictive use of the term “uncertainty” has been debated and developed during the last decade in the risk assessment community ([1][2][3]). In the literature it is also referred to as epistemic uncertainty and reducible uncertainty [4].

The need for expressing uncertainty in simulation models is widely recognized (see for example [5][6]). In fact, there has been a great deal of work done on this subject in high risk areas, for example in hydrology ([7][8][9][10][11]), climate change[12] and whale management ([13][14][15]). The emphasis in uncertainty has been directed toward assessing simulation model input parameter uncertainty. And the vast majority of this work has used probabilistic methods to represent sources of variability or uncertainty and then sampling methods, such as Monte Carlo sampling, to propagate the sources [4], while simulation output uncertainty is rarely discussed. Although in some literatures, simulation output uncertainty is discussed together with input uncertainty, conclusion regarding simulation output uncertainty is often drawn from input uncertainty based on some mathematical models (such as Bayesian Analysis and Bayesian Belief Networks). In many times we need the process of dealing with simulation output uncertainty without taking much input uncertainty into consideration.

Our focus in this paper is to construct a simple model to analysis simulation output uncertainty for stochastic simulation systems. The model, or the algorithm described in this paper will not depend much on the input uncertainty and can be used in many stochastic simulation systems.

## 2 A Brief Review of Simulation Uncertainty Analysis

### 2.1 Review of Simulation Uncertainty Analysis

Recently proposed methods regarding uncertainty include Delta-Method approaches, Resampling Method and Bayesian Method. Delta-Method Approaches Started with [16] and continued with [17][18][19]. The framework given by [16] has been adopted by several authors including [20][21]. For an introduction and overview of Resampling methods in simulation, see [22][23]. Barton and Schruben ([24]) proposed two resampling methods for accounting for input uncertainty. They used empirical distribution functions to model the distribution functions of independent input random variables.

There has been a fair amount of recent interest in Bayesian methods for simulation input analysis ([22][23][25][26][27][28][29]). The idea of applying Bayesian techniques to simulation analysis is not new, however, an earlier reference can be found in [25]. The overall philosophy behind these methods is to place a prior distribution on the input models and parameters of a simulation, update the prior distribution to a posterior distribution based on available data, and only then run a simulation experiment. The Bayesian framework is an elegant one that enables a clean answer to many vexing questions. There are, however, several issues that deserve further attention. Perhaps the key issue is that of computational efficiency. It can be quite difficult to compute the posterior distribution in general, so that one often has to

resort to computational devices like Markov chain Monte Carlo methods or importance sampling.

A simple and straightforward tool that is often used is sensitivity analysis. A sensitivity analysis (e.g., [30][31]) is performed by varying the input distributions and parameters in some manner, and observing the changes in the output. This is often done in a somewhat haphazard way, although there are benefits to formalizing the approach using design of experiments and/or regression approaches [32].

One can quite reasonably argue that the methods we review here are all for input model parameter uncertainty analysis, while the simulation output uncertainty is seldom discussed. In fact, in most of the cases, simulation output uncertainty is deduced based on the input uncertainty. However, in a simulation system, there are so many parameters in the simulation model. It is impossible to assess uncertainty for all parameters. Also in a stochastic simulation system, there are many temporarily random parameters embedded in the system and possibly cannot be assessed by the above mentioned methods. We need a simple method to assess uncertainty for simulation output instead of being deduced from input uncertainty.

In Section 3, we will propose a simple and straightforward method to assess simulation output uncertainty based on a new method, Bayesian Melding, which will be discussed in the next subsection.

## 2.2 Bayesian Melding

Bayesian melding was proposed by [13] and [14] as a way of putting the analysis of simulation models on a solid statistical basis. The basic idea is to combine all the available evidence about model inputs and model outputs in a coherent Bayesian way, to yield a Bayesian posterior distribution of the quantities of interest. The method was developed initially for deterministic simulation models.

We denote the collection of model inputs about which there is uncertainty by  $\Theta$ ; collection of model outputs about which we have some observed information by  $\Phi$ . In the case of a deterministic system, we denote the mapping function that produces  $\Phi$  by  $M_\Phi$ , so that  $\Phi = M_\Phi(\Theta)$ . The first step of the method is to encode the available information about model inputs and outputs in terms of probability distributions. We represent our information about the inputs,  $\Theta$ , by a prior probability distribution,  $q(\Theta)$ . We specify a conditional probability distribution of the data  $y$  given the outputs  $\Phi$ , and this yields a likelihood for the outputs

$$L(\Phi) = \text{Pr ob}(y | \Phi). \tag{1}$$

Because  $\Phi = M_\Phi(\Theta)$ , (1) yields a likelihood for the inputs also, since

$$L(\Theta) = \text{Pr ob}(y | M_\Phi(\Theta)). \tag{2}$$

We thus have a prior,  $q(\Theta)$  and a likelihood,  $L(\Theta)$ , both defined in terms of the inputs. It follows from Bayes's theorem that we have a posterior distribution of the inputs given all the available information, namely

$$\pi(\Theta) \propto q(\Theta)L(\Theta). \tag{3}$$

In words, the posterior density is proportional to the prior density times the likelihood. The constant of proportionality is defined so that  $\pi(\Theta)$  is a probability density, i.e. so that it integrates to 1. In principle, (3) yields a full posterior distribution,  $\pi(\Phi)$ . This combines all the available relevant information, and so provides a comprehensive basis for risk assessment and decision-making. It works as follows:

- (1). Draw a sample  $\{ \Theta_1, \dots, \Theta_I \}$  of values of the inputs from the prior distribution  $q(\Theta)$ .
- (2). Obtain  $\{ \Phi_1, \dots, \Phi_I \}$  where  $\Phi_i = M_\Phi(\Theta_i)$ .
- (3). Compute weights  $w_i = L(\Phi_i)$ .
- (4). The approximate posterior distribution of simulation output has values  $\{ \Phi_1, \dots, \Phi_I \}$  where  $\Phi_i = M_\Phi(\Theta_i)$  and probabilities proportional to  $\{ w_1, \dots, w_I \}$ .

The models that we are dealing with here are stochastic models, which may use random numbers, run with different seeds and return different results. To include this source of uncertainty in the framework, Sevcikova H. et al. [15] modified the above procedure as follows:

- 1S. As before, draw a sample  $\{ \Theta_1, \dots, \Theta_I \}$  of values of the inputs from the prior distribution  $q(\Theta)$ .
- 2S. For each  $\Theta_i$ , run the model  $J$  times with different seeds to obtain  $\Phi_{ij}, j=1, \dots, J$ .
- 3S. Compute weights  $w_i = L(\Phi_i)$ . We then get an approximate posterior distribution of inputs with values  $\{ \Theta_1, \dots, \Theta_I \}$  and probabilities proportional to  $\{ \bar{w}_i : i=1, \dots, I \}$  where  $\bar{w}_i = \frac{1}{J} \sum_{j=1}^J w_{ij}$ .
- 4S. The approximate posterior distribution of  $\Phi$  now has  $I \times J$  values  $\Phi_{ij} = M_\Phi(\Theta_i)$ , with weights  $w_{ij}$ .

The revised Bayesian Melding method provides us a very useful method to evaluate simulation output uncertainty directly without considering input uncertainty too much for the uncertainty information used in the algorithm is based on simulation output. Two key problems are: (1) the determination of probability density function of simulation output and (2) the calculation of weights. For the input parameter samples, since in many times, we know their prior probability distribution functions (pdf) before, or they can be acquired from other methods, we can generate their values by such method as Monte Carlo methods according to their prior pdf, and then sample from the parameter value set to feed into the simulation system. Section 3 will give the process in details.



### 3 Simulation Output Uncertainty Assessing Based on Bayesian Melding Method and Sensitivity Analysis

#### 3.1 Basic Ideas

The basic idea of our algorithm to assess simulation output uncertainty is to combine Sensitivity analysis and Bayesian Melding method to construct a simple and straightforward framework instead of listing so many mathematical equations. The reasons are apparent. On one hand, we hope the uncertainty analysis model could be used by users rather than discussed by scholars. On the other hand, in many stochastic simulation systems, the prior distributions of some important parameters are known or can be deduced by other methods. We can use these prior distributions to generate parameter value sets, then sample from these datasets, run simulations with sampled datasets and finally, analysis the simulation output uncertainty from the specified outputs. Since every simulation may convey information from output population and it is impossible for user to run simulation endless times, we define that the final output uncertainty be the weight sum of every simulation output.

Another problem is the form of probability distribution function of simulation output. Since it is difficult to acquire exact form of probability distribution function for simulation output in a stochastic simulation system, we can take the assumption that the simulation output is a normal distribution. In the following subsection, we will take advantage of this assumption to calculate the final pdf of simulation output.

#### 3.2 Algorithm

The algorithm we propose to find the pdf of stochastic simulation output is described below.

Step 1. Calculate the prior distribution function  $q(\theta_i)$  for parameter  $\theta_i$  ( $q(\theta_i)$  maybe known before or can be deduced from other methods);

Step 2. Generate Parameter values: for each parameter  $\theta_i$ , generate  $M_i$  values according to prior distribution function  $q(\theta_i)$  to form a parameter value set  $\Lambda_i$ ;

Step 3. Sample: for each parameter  $\theta_i$ , draw samples from parameter value set  $\Lambda_i$ ; the total sample number is  $I$ . We then have  $I$  input parameter combinations  $\{\Theta_1, \dots, \Theta_I\}$ .

Step 4. Simulation: for each combination of input parameters  $\Theta_i$ , run simulation  $J$  times (in order to reduce the simulation output error incurred by random errors), we then get simulation output  $\Phi_{ij}$  ( $i=1,2,\dots,I; j=1,2,\dots,J$ ). The simulation output may be  $K$  dimensions.

Step 5. Assume the pdf form of random variable  $\Phi|\Theta$  (simulation output), estimate the parameter value in the pdf of simulation output, we then get the pdf of  $\Phi|\Theta$ , namely,  $p(\Phi|\Theta)$ .

Step 6. Compute weight  $w_{ik}$  ( $i \in I, k \in K$ ).  $w_{ik}$  satisfies Equation (4) and (5),

$$w_{ik} \propto p(\Phi_k | \Theta_i). \quad (4)$$

$$\sum_{i=1}^I w_{ik} = 1. \tag{5}$$

Step 7. Calculate the approximate posterior distribution of simulation output  $\Phi$  : for each simulation output  $\Phi_k$ ,

$$\pi(\Phi_k) = \sum_{i=1}^I w_{ik} p(\Phi_k | \Theta_i). \tag{6}$$

From the description of the algorithm, we can see that the final simulation output uncertainty is deemed as the weight sum of the multiple simulation output uncertainty, and the weights are proportional to the probability of the simulation output. The advantages of the revised algorithm are: (1) The pdf form of simulation output is known before assessment and its parameters can be estimated from simulation outputs; (2) Compared with other simulation output uncertainty method, such method has a good application prospect for it provides a simple and straightforward framework to assess simulation output uncertainty without considering input model parameter uncertainty too much. However, there are two key problems left unresolved in the algorithm: (1) the form of pdf for simulation output and the estimation of parameters in the pdf; (2) the calculation of the weights. In the following subsections, we will discuss them thoroughly.

### 3.3 Estimation of Parameters

According to Bayes theory,  $\Phi | \Theta$  is a random variable. We now assume the random variable takes the form of pdf as normal distribution (parameters unknown). We have Equation (7),

$$\Phi_{ijk} = u_{ik} + \delta_{ijk}. \tag{7}$$

in which  $\delta_{ijk} \stackrel{iid}{\sim} N(0, \sigma_{ik}^2)$  ( $i \in I, j \in J, k \in K$ ).  $\mu_{ik}$  denotes the mean of  $\Phi_{ijk}$  (please be noted that for parameter combination  $\Theta_i$ , we run simulation  $J$  times);  $\delta_{ijk}$  is model error. Since we have known the pdf form of  $\Phi | \Theta$ , we need to estimate their parameters. The estimation of these parameters is given by Equation (8)-(11).

$$\hat{\mu}_k = \frac{1}{IJ} \sum_{i=1}^I \sum_{j=1}^J \Phi_{ijk}. \tag{8}$$

$$\hat{\mu}_{ik} = \frac{1}{J} \sum_{j=1}^J \Phi_{ijk}. \tag{9}$$

$$\hat{\sigma}_{ik}^2 = \frac{1}{J} \sum_j (\Phi_{ijk} - \hat{\mu}_{ik})^2. \tag{10}$$

$$\hat{\sigma}_k^2 = \frac{1}{I} \sum_i (\hat{\mu}_{ik} - \hat{\mu}_k)^2. \tag{11}$$

Therefore,  $\Phi_{ik} | \Theta_i \sim N(\hat{\mu}_{ik}, \hat{\sigma}_{ik}^2)$ , i.e.,

$$p(\Phi_{ik} | \Theta_i) = \frac{1}{\sqrt{2\pi} \hat{\sigma}_{ik}} \exp[-\frac{1/2(y - \hat{\mu}_{ik})^2}{\hat{\sigma}_{ik}^2}]. \tag{12}$$

The final simulation output uncertainty is deemed as the weight sum of the multiple simulation output uncertainty, namely,

$$p(\Phi_k | \Theta) = \sum_{i=1}^I w_{ik} p(\Phi_{ik} | \Theta_i) = \sum_{i=1}^I (w_{ik} \frac{1}{\sqrt{2\pi\hat{\sigma}_{ik}}} \exp[-\frac{1/2(y - \hat{\mu}_{ik})^2}{\hat{\sigma}_{ik}^2}]). \tag{13}$$

### 3.4 Weights

We define the final simulation output uncertainty pdf as the weight sum of the pdf for each simulation output, namely,

$$p(\Phi_k | \Theta) = \sum_{i=1}^I w_{ik} p(\Phi_{ik} | \Theta_i) = \sum_{i=1}^I (w_{ik} \frac{1}{\sqrt{2\pi\sigma_{ik}}} \exp[-\frac{1/2(y - \hat{u}_{ik})^2}{\sigma_{ik}^2}]). \tag{14}$$

In Step 6 of the Algorithm, we assume the weight  $w_{ik}$  ( $i \in I, k \in K$ ) be proportional to  $p(\Phi_{ik} | \Theta_i)$ . To calculate  $w_{ik}$ , the first thing we have to do is to calculate  $p(\Phi_{ik} | \Theta_i)$  for a given  $\Phi_{ik}$ . This is done by Equation (15).

$$p(\Phi_{ik} | \Theta_i) = \begin{cases} 1 - \int_{\hat{\mu}_{ik}}^{\hat{\mu}_{ik} + 2\Delta u_{ik}} p'(\Phi_k | \Theta) dy & \text{if } \hat{u}_k \geq \hat{\mu}_{ik} \\ 1 - \int_{\hat{\mu}_{ik} - 2\Delta u_{ik}}^{\hat{\mu}_{ik}} p'(\Phi_k | \Theta) dy & \text{else} \end{cases} \tag{15}$$

in which  $\Delta u_{ik} = |\hat{u}_k - \hat{\mu}_{ik}|$  and  $p'(\Phi_k | \Theta) = \frac{1}{\sqrt{2\pi\hat{\sigma}_k}} \exp[-\frac{1/2(y - \hat{\mu}_k)^2}{\hat{\sigma}_k^2}]$ . We can see that the more  $\hat{\mu}_{ik}$  is near  $\hat{u}_k$ , the more  $p(\Phi_{ik} | \Theta_i)$  will be and vice versa. The weight for each simulation output can be determined by Equation (16)

$$w_{ik} = \frac{p(\Phi_{ik} | \Theta_i)}{\sum_{i=1}^I p(\Phi_{ik} | \Theta_i)}. \tag{16}$$

## 4 Case Study

For a better understanding of this method, we propose a case study in this section. The stochastic simulation system is for traffic system, in which, simulation modeling is a very important tool to study. For more details about the stochastic simulation system, please refer [33].

In [33], we can find that for all simulation tests, the arriving rate of vehicles is a very important parameter in studying traffic simulation. We therefore take this parameter as the only random parameter to study and take the one-lane segment simulation test described in [33] as the subject test.

Since the arriving rate distribution of vehicles is a Poisson distribution with a mathematic expectation of 300 vehicles per 15 minutes. We firstly used matlab(6.5) function *poissrnd* to generate 10 values {300, 293, 304, 263, 293, 297, 278, 292, 302, 321}. The expectation of them is 300. We then drew 5 samples ( $I = 5$ ) from them.

The sampled dataset was {297, 321,300,293,304}. The sampled parameters were fed into the simulation system to find the simulation output. For each parameter, we ran simulation 3 times ( $J = 3$ ). Table 1 gives the final output for each simulation. We provided here speed as the only output variable ( $K = 1$ ). Table 1 also gives the estimation of  $\hat{\mu}_{ik}, \hat{\sigma}_{ik}^2, \hat{\mu}_k$  and  $\hat{\sigma}_k^2$ .

**Table 1.** Simulation Output and Parameter Estimation

$i$	Arriving Rate	$j$	Speed ( $K=1$ )	$\hat{\mu}_{ik}$	$\hat{\sigma}_{ik}^2$	$\hat{\mu}_k$	$\hat{\sigma}_k^2$
1	297	1	72.3	72.4	0.68		
		2	71.5				
		3	73.5				
2	321	1	70.4	70.7	0.86		
		2	72.0				
		3	69.8				
3	300	1	72.0	73.1	0.75	72.24	1.44
		2	74.1				
		3	73.3				
4	293	1	74.5	73.9	0.24		
		2	73.3				
		3	74.0				
5	304	1	71.8	71.1	0.23		
		2	70.9				
		3	70.7				

Equations (17)-(21) list pdf of simulation output based on Table 1. Equation (22) is for  $p'(Speed | \Theta)$ , which will be used to calculate weights. And the probabilities calculated for each simulation output through Equation (22) and Equation (15) are listed in Table 2. Table 2 also gives the final result of weights through Equation (16) and the calculated probabilities for each simulation output.

$$p(Speed | \Theta_1) = \frac{1}{\sqrt{1.36\pi}} \exp\left[-\frac{1/2(y - 72.4)^2}{0.68}\right]. \tag{17}$$

$$p(Speed | \Theta_2) = \frac{1}{\sqrt{1.72\pi}} \exp\left[-\frac{1/2(y - 70.7)^2}{0.86}\right]. \tag{18}$$

$$p(Speed | \Theta_3) = \frac{1}{\sqrt{1.5\pi}} \exp\left[-\frac{1/2(y - 73.1)^2}{0.75}\right]. \tag{19}$$

$$p(Speed | \Theta_4) = \frac{1}{\sqrt{0.48\pi}} \exp\left[-\frac{1/2(y - 73.9)^2}{0.24}\right]. \tag{20}$$

$$p(Speed | \Theta_5) = \frac{1}{\sqrt{0.46\pi}} \exp\left[-\frac{1/2(y - 71.1)^2}{0.23}\right]. \tag{21}$$

$$p'(Speed | \Theta) = \frac{1}{\sqrt{2.4\pi}} \exp\left[-\frac{1/2(y - 72.24)^2}{1.44}\right]. \tag{22}$$

**Table 2.** Calculation of Probability and Weights

I	1	2	3	4	5
Probability	0.8966	0.1995	0.4736	0.1667	0.3421
Weight	0.4314	0.0960	0.2279	0.0802	0.1645

(Note: The calculation of probability is performed by Matlab 6.5 function *quadl*)

As a result, the final simulation output probability distribution functions can be described as:

$$p(Speed | \Theta) = \sum_{i=1}^I w_{ik} p(Speed | \Theta_i) = \frac{0.4314}{\sqrt{1.36\pi}} \exp\left[-\frac{1/2(y - 72.4)^2}{0.68}\right] + \frac{0.096}{\sqrt{1.72\pi}} \exp\left[-\frac{1/2(y - 70.7)^2}{0.86}\right] + \frac{0.2279}{\sqrt{1.5\pi}} \exp\left[-\frac{1/2(y - 73.1)^2}{0.75}\right] + \frac{0.0802}{\sqrt{0.48\pi}} \exp\left[-\frac{1/2(y - 73.9)^2}{0.24}\right] + \frac{0.1645}{\sqrt{0.46\pi}} \exp\left[-\frac{1/2(y - 71.1)^2}{0.23}\right] \tag{23}$$

## 5 Conclusion and Remark

One can quite reasonably argue that so long as the simulation user is aware of potential model errors due to input model uncertainty, interprets the simulation output accordingly, and conducts sensitivity and/or uncertainty analyses all is well. However, the problem is that the typical simulation user is not particularly proficient in statistics, and so is unlikely to be aware of appropriate sensitivity and/or uncertainty analyses. This suggests the need for a transparent, statistically valid, implementable and efficient method for simulation uncertainty [32].

However, in the recent proposed uncertainty analysis methods regarding stochastic simulation systems, most of them deal with the model input uncertainty rather than simulation output. In this paper, we proposed a simple and straightforward framework to assess stochastic simulation output uncertainty based on Bayesian Melding. To make the analysis, we need only to do two things (1) to estimate parameters in the simulation output probability distribution function and (2) to calculate weight for each simulation. Both of them are discussed in detail in this paper. We also provided a case study in the paper for the purpose of better understanding. Although the assumption that the simulation output is a normal distribution is somewhat a bit arbitrary (one of the typical ways is to make hypothesis test), when compared with other complicated methods, the most apparent advantages are simple and straightforward. We may sometime need some complicated mathematical equations to get parameter prior pdf in order to generate proper parameter value sets. And this can be done through such method as MCMC. Besides, it is impossible to make uncertainty analysis for all input parameters. A pre-sensitivity analysis is needed to find the most sensitive and/or important parameters when too many parameters participate the uncertainty analysis.

**Acknowledgments.** The research is supported by the Beijing Nature Science Foundation (No. 8033015), NSFC (No. 70571076, 40471111) and MOST (No. 2006AA12Z215).

## References

1. Morgan, M.G., Henrion, M.: *Uncertainty: A guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, 1st edn. Cambridge University Press, New York (1990)
2. Beck, M.B.: *Water Quality Modeling: A Review of the Analysis of Uncertainty*. *Water Resources Research* 23(8), 1393–1442 (1987)
3. Bogen, K.T., Spear, R.C.: *Integrating Uncertainty and Interindividual Variability in Environmental Risk Assessment*. *Risk Analysis* 7(4), 427–436 (1987)
4. Oberkampf, W.L., et al.: *Estimation of Total Uncertainty in Modeling and Simulation*. SANDIA REPORT, SAND2000-0824, Unlimited Release (2000)
5. Refsgaard, J.C., Henriksen, H.J.: *Modelling guidelines – terminology and guiding principles*. *Advances in Water Resources* 27(1), 71–82 (2004)
6. Van Asselt, M.B.A., Rotmans, J.: *Uncertainty in integrated assessment modelling – from positivism to pluralism*. *Climatic Change*, 75–105 (2002)
7. Beven, K.: *Rainfall-runoff Modelling: The Primer*. John Wiley & Sons, Chichester (2000)
8. Beven, K., Binley, A.: *The future of distributed models: model calibration and uncertainty prediction*. *Hydrological Processes* 6(3), 279–298 (1992)
9. Christensen, S.: *A synthetic groundwater modelling study of the accuracy of GLUE uncertainty intervals*. *Nordic Hydrology* 35(1), 45–59 (2003)
10. Neuman, S.P.: *Maximum likelihood Bayesian averaging of uncertain model predictions*. *Stochastic Environmental Research and Risk Assessment* 17(5), 291–305 (2003)
11. Korving, H., Van Noordwijk, J.M., Van Gelder, P.H.A.J.M., Parkhi, R.S.: *Coping with uncertainty in sewer system rehabilitation*. In: van Gelder, B. (eds.) *Safety and Reliability*. Swets & Zeitlinger, Lisse (2003)
12. Casman, E.A., Morgan, M.G., Dowlatabadi, H.: *Mixed levels of uncertainty in complex policy models*. *Risk Analysis* 19(1), 33–42 (1999)
13. Raftery, A.E., Givens, G.H., Zeh, J.E.: *Inference from a deterministic population dynamics model for bowhead whales (with discussion)*. *Journal of the American Statistical Association*, 402–416 (1995)
14. Poole, D., Raftery, A.E.: *Inference for deterministic simulation models: the Bayesian melding approach*. *Journal of the American Statistical Association*, 1244–1255 (2000)
15. Sevcikova, H., et al.: *Assessing uncertainty in urban simulations using Bayesian Melding*. *Transport. Res. Part B* (2006), doi:10.1016/j.trb.2006.11.001
16. Cheng, R.C.H.: *Selecting input models*. In: Tew, J.D., Manivannan, S., Sadowski, D.A., Seila, A.F. (eds.) *Proceedings of the 1994 Winter Simulation Conference*, pp. 184–191. IEEE, Piscataway, NJ (1994)
17. Cheng, R.C.H., Holland, W.: *Sensitivity of computer simulation experiments to errors in input data*. *Journal of Statistical Computation and Simulation* 57, 219–241 (1997)
18. Cheng, R.C.H., Holland, W.: *Two-point methods for assessing variability in simulation output*. *Journal of Statistical Computation and Simulation*, 183–205 (1998)
19. Cheng, R.C.H., Holland, W.: *Calculation of confidence intervals for simulation output*. submitted for publication, 2003

20. Zouaoui, F., Wilson, J.R.: Accounting for input model and parameter uncertainty in simulation. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (eds.) *Proceedings of the 2001 Winter Simulation Conference*, pp. 290–299. IEEE, Piscataway, NJ (2001)
21. Zouaoui, F., Wilson, J.R.: Accounting for parameter uncertainty in simulation input modeling. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (eds.) *Proceedings of the 2001 Winter Simulation Conference*, pp. 354–363. IEEE, Piscataway, NJ (2001)
22. Cheng, R.C.H.: Analysis of simulation output by resampling. *International Journal of Simulation: Systems, Science & Technology* 1, 51–58 (2000)
23. Cheng, R.C.H.: Analysis of simulation experiments by bootstrap resampling. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (eds.) *Proceedings of the 2001 Winter Simulation Conference*, pp. 179–186. IEEE, Piscataway, NJ (2001)
24. Barton, R.R., Schruben, L.W.: Uniform and bootstrap resampling of empirical distributions. In: Evans, G.W., Mollaghasemi, M., Russell, E.C., Biles, W.E. (eds.) *Proceedings of the 1993 Winter Simulation Conference*, pp. 503–508. IEEE, Piscataway, NJ (1993)
25. Chick, S.E.: Bayesian analysis for simulation input and output. In: Andradóttir, S., Healy, K.J., Withers, D.H., Nelson, B.L. (eds.) *Proceedings of the 1997 Winter Simulation Conference*, pp. 253–260. IEEE, Piscataway, NJ (1997)
26. Chick, S.E.: Steps to implement Bayesian input distribution selection. In: Farrington, P.A., Black Nembhard, H., Sturrock, D.T., Evans, G.W. (eds.) *Proceedings of the 1999 Winter Simulation Conference*, pp. 317–324. IEEE, Piscataway, NJ (1999)
27. Chick, S.E.: Bayesian methods for simulation. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (eds.) *Proceedings of the 2000 Winter Simulation Conference*, pp. 109–118. IEEE, Piscataway, NJ (2000)
28. Chick, S.E.: Input distribution selection for simulation experiments: accounting for input uncertainty. *Operations Research*, 744–758 (2001)
29. Chick, S.E., Ng, S.H.: Joint criterion for factor identification and parameter estimation. In: Yücesan, E., Chen, C.-H., Snowdon, J.L., Charnes, J.M. (eds.) *Proceedings of the 2002 Winter Simulation Conference*, pp. 400–406. IEEE, Piscataway, NJ (2002)
30. Kleijnen, J.P.C.: Sensitivity analysis versus uncertainty analysis: when to use what? In: Grasman, J., van Straten, G. (eds.) *Predictability and Nonlinear Modeling in Natural Sciences and Economics*, Kluwer Academic, Dordrecht (1994)
31. Kleijnen, J.P.C.: Five-stage procedure for the evaluation of simulation models through statistical techniques. In: Charnes, J.M., Morrice, D.J., Brunner, D.T., Swain, J.J. (eds.) *Proceedings of the 1996 Winter Simulation Conference*, pp. 248–254. IEEE, Piscataway, NJ (1996)
32. Shane, G.: Henderson: Input Model Uncertainty: Why Do We Care And What Should We Do About It? In: *Proceedings of the 2003 Winter Simulation Conference* (2003)
33. Sun, T., Wang, J.: A traffic cellular automata model based on road network grids and its spatial & temporal resolution's influences on simulation. *Simulation Modelling Practice and Theory* (2007), <http://dx.doi.org/10.1016/j.simpat.2007.04.010>

# An Empirically Terminological Point of View on Agentism in the Artificial

C.T.A. Schmidt

Le Mans University

LIUM

52, Rue des Docteurs Calmette et Guérin,

53020 Laval France

Colin.Schmidt@univ-lemans.fr

**Abstract.** Many endeavours in Artificial Intelligence work towards recreating the dialogical capabilities of humans in machines, robots, "creatures", in short information processing systems. This original goal in AI has been left to the wayside by many in order to produce Artificial Life entities in a futuristic vision of 'life-as-it-could-be'; scientists that have not 'abandoned ship' confirm the difficulty of reaching the *summum* of AI research. This means the importance of language generation and understanding components have been reduced. Are the pragmatics of language use too difficult to deal with? According to Shapiro and Rapaport (1991), "the quintessential *natural-language competence* task is interactive dialogue". Man-made entities are not functional in dialoguing with humans. The benefits of re-establishing a "proper" relational stance in the Artificial Sciences are twofold, namely, *a.* to better understand the communication difficulties encountered, and *b.* to bring enhanced meaning to the goals of building artificial agents. Point *a* has consequences for *b* in that it will change the very goals of scientists working on social and conversational agents. In the literature, the *notion* of agent proves unsuitable for the specification of any higher-order communication tasks; a Tower of Babel problem exists with regards to the very definition of "agent" between Scientists and Philosophers. In the present article, I eliminate the nebulosity currently contouring agency's terminology with a goal to improving understanding when speaking about entities that can mean.

**Keywords:** Intentionality, Communication, Agency, Technological artefacts, Epistemology.

## 1 Introduction

M. Bedau *et al.* give a list of "open problems" for Artificial Life (2000 p. 364-365) meaning they are not yet solved or could never be solved; one must be wary though, "technology is on the boom" say many scientists. But technology that blends truth and falsehood cannot make sincere headway. The type of terminology used in Artificial Life leads to confusion because, in many cases, the scientists of the artificial are referring to *notions* that have not been "pinned down" yet. If the very terminology



in a field also constitutes an open problem (of definition), how can the scientist attack the field's inherent problems? In this article, *I will conduct a philosophical investigation into this problem that has its negative effects at the practical level of scientists using language.*

And so a Tower of Babel stands proudly at the intersection of Computer Science, Philosophy and Artificial Life with respect to a key topic in the literature: agentism (no political connotations intended). Is the term "agent" still operational? Not if you find yourself caught up in this intersection, as is the case for many ontologists. If one were to focus on the notion rather than on the word, it may be possible to reduce misunderstandings somewhat. After all, the essence of employing natural language is a way to master the intangibility of thought or, as Shapiro & Rapaport so concisely put in their "Minds and Models" article (1991), to use our "ability to discuss the not-here and the not-now". Would playing with the term "agent" and exploring its possible meanings tear down the Tower that hinders effective communication between scientists?

Agents in fact are suffering a severe identity crisis.

In Philosophy, D. Davidson wrote an article entitled, quite simply, "Agency" (1971). Exploring universal notions seems to be an endeavour left to the wayside in today's gallop towards 'fast technology'. Are they no longer relevant? Why were they ever relevant in the past? In designing technological objects having the function of interacting with humans, one of the metaphors used is that of interpersonal human communication. In order to do this, the theoretical basis supporting the metaphor must explain the phenomena singled out (i.e. communication as a relation-based activity). Explaining human activity often calls upon notions like intentionality<sup>1</sup> and agency — the quality of being able to act. D. Elgesem (1997) points to Davidson's claim that "the mark of agency is intentionality *under some description*" (p. 3). In this article, I will focus on the meaning of the words I emphasise here in the context of building socially or conversationally endowed agents.

Clarification involving formal ontology is necessary here in order to banish the ambiguity that wreaks havoc upon the reading of agency-related words. I, for one, do not always understand when reading about agency in technical fields; this is most likely because I ('universally') see "agents" to be *static things* or *evolving processes* in information systems (according to B. Smith, *substantialists* and *fluxists* each respectively only see one of these options a possible<sup>2</sup>). I think it is clear now to the reader that in my approach, I mean to analyse discourse used in speaking about the design of information systems from an experimental and pragmatic point of view.

## 2 For the Rediversification of a Concept

Whether one is speaking about screen-animated characters, humanoids, electronic pets, interfaces, a human's implication in a collective endeavour or his position in society, one often needs to employ the term "agent". And what if using this term were not appropriate? My intention is to evaluate the use of such terminology in the areas

---

<sup>1</sup> Intentionality refers to beliefs, desires, intentions, etc. Cf. J. Searle's 1983 Intentionality for a full list.

<sup>2</sup> Cf. Smith B. (draft), "Ontology and Information Science", p. 3.

of scientific study that have a steadfast footing in both the natural and the artificial. While I believe that when employing the term "agent" for speaking *either* about natural entities *or* artificial ones we can understand each other most of the time, I am convinced that when one mixes the application fields we complicate the matter and the chances of coming to true understanding plummet.

In addressing the technical issues, one cannot wield the term "agent" savagely. Skills and know-how vary immensely from agent to agent. In fact, using the *notion* of agent to conceptually render *generic* the various participants —humans and artefacts— in multi-agent systems is something that should have been avoided. Who started this? It opens the floodgates to ambiguity.

It is difficult to understand the topic of discussion if almost everything in a system is referred to as an "agent". We lose specification detail by doing so.

People and animated artefacts produce acts which then belong to the past; in order to assign full meaning to an act, one appreciates having an idea of the nature of the entity propagating it (i.e. human, half-human, animal, non-human, etc.). This may still be the case as, when people see artefacts, they still expect to attribute any intentionality the entity may *display* to its designer. For making machinery (seem) intelligent, D. Dennett (1996) and J. Zlatev (2001) put forth the importance of the learning-by-doing option over the simple programming-in method commonly used. The way I see it, simply not knowing the origin of the "species" one is looking at could pose a problem for the average 'non-techie' trying to interpret Artificial Life type creatures.

Lending "agent stuffs" (matter) ways of expressing social drive and ways of reproducing and refining the physical manifestations of human or animal states has become popular all the same. Consider the heavily financed programmes in A-Life, Robotics and Computer Science information systems in Japan<sup>3</sup> or just look at the state-of-the-art projects in R. Brooks' team at MIT<sup>4</sup>. Many of these specialists, and others working in field closely related to Artificial Life, use the term "*agent*" when speaking about their very visible creations within the framework of a larger entity, an interactive system containing other agents (humans), or even society itself (*cf.* concepts like *collective* or '*swarm*' *intelligence*, *distributed cognition* in the literature). But is the use of the term "*agent*" restrained to an application upon physical entities?

From the title, the reader might get the impression that I have a special person I would like she or him to meet, but this is really about a *non-person*. The *notion* of agent has been a handy notion at the intersection of Artificial Intelligence, Artificial Life and human Society for many years now. Agents represent the artificial *and*

<sup>3</sup> The private sector is booming too. NEC has built a personal robot called "Papero" that walks and talks following your face in conversation and allegedly speaks differently with different interlocutors. Similarly, Toshiba produces the "Aprialpha" Robodex 2003 and "Ifbot" by Business Design Laboratory Company is like "Papero". Fujitsu has developed a mobile phone-controlled robot for home use called MARON-1 : it can be remotely controlled by mobile phone to operate home electronic appliances or monitor household security.

<sup>4</sup> Cf. the web pages (i.e. <http://www.csail.mit.edu/research/abstracts/abstracts03/robotics/robotics.html> or <http://www.csail.mit.edu/research/abstracts/abstracts03/artificial-life/artificial-life.html>) of the recently established *Computer Science and Artificial Intelligence Laboratory* (CSAIL).

human participants within current systems<sup>5</sup>. The notion of agent is thus used to isolate similar actions, to say that they are performed by a “*who*” (human) or an “*it*” (non-human machine) without having to explain the *identity* of the “*who*” or the *nature* of the “*it*” in an in-depth manner. Sociologists, psychologists and philosophers have of course been more intrigued by the identity of humans (or other related concepts) than specialists in Biology, Computer Scientists and Artificial Intelligence have; this latter group of specialists on the other hand have given greater attention to the characteristics of some non-human entities having a “real function” than specialists of the social or human sciences have. This division—which incidentally crops up throughout all test-beds descending both from the humanities and the ‘hard’ sciences—is almost neither here nor there when one employs the word “agent” to put the accent on the *performing* rather than on the person or object that performs the action. But can one blend two divergent points of view with a mere word?

The aim of this article is to repel by argument the coming of agentism to the Information Technologies in order to improve the *relational foundations* for the future developments in the computerisation process, especially when it comes to questions of communication. I shall base my refutation of the notion of agent on a practical enquiry into its use. For instance, why has this notion increasingly become so important to the process of producing encounters between machines and members of human society over the last twenty years?

### 3 Mind, Communication and Explanation

The uproar of physicalism in the study of cognitive systems study has been made possible by advances in the neuro-sciences. This has not been sufficient enough to replace exploring mind approaches with explaining brain ones. Any human capacities and activities, like communication, that presuppose using the notion of mind for their functional explanation require a simplification factor. The simplification factor that agentism has been seen to augment is the major reason it has come into being; a certain homogeneity in the reading of a system comes from applying it to all potential actors in the system, whatever their nature may be (of course this implies the *under-description* of the actors).

It would seem that the ‘slide’ from well-defined actors towards actions has come out of the necessity (and the ease) to reduce the attention given to the description of the characteristics of objects, as Science has usually done in the past. Some of computing’s products, as contributions to Artificial Life research, seem to lie at centre ground between soul-bearers (persons) and the world of tangible things (objects including machines). Forcing humans to focus on actions rather than on the *who*’s or *it*’s involved within a socio-technological device—systems that include humans and machines—enables designers and people working in (more or less) strong Artificial Intelligence to short-circuit the explanatory

---

<sup>5</sup> Today we commonly find all sorts of these notions expressed in fields having to do with computerising society, i.e. information agents, domain agents, task-handling agents, user agents etc. In his 1997 book (in French), French Professor D. Vernant establishes a short but all-encompassing list of agents including R. Brooks’ robotic creatures, animal organisms and humans, cf. the chapter entitled “From Action to Communication”.

differences between *who*'s and *it*'s, especially when speaking about users and the machine-like artefacts designed for them. In fact, shifting to agentism allows scientists to merge these two explanatory vectors, that of artificial and natural entities (materials, cognitive brain/mind processes, etc.), into one single explanation for both. One single description of the performer, one single definition of agent. The reason for wanting to delve into 'manipulations' of human discernment is to render more plausible the automating of certain tasks that have always, until recently, been performed by *who*'s and not *it*'s.

Some of the tasks that have been handed over to robots and other machines have been done so with reason as they involve tedious repetitive activities (handling booking information for events, hotels etc., explaining transportation schedules, using popular equations like temperature or currency conversions, conducting asynchronous communication link-ups...). But in situations in which the enquirer does not come 'equipped' with a fully formulated question (and ask it in the proper context), automation of providing correct information proves difficult. It shows the difficulties speakers encounter in the unintelligible dialogues that result all too often in person-machine situations. Two examples of this could be when one is looking for a book without really having a clear idea of the subject area of what one wants to read or when one is learning how to cope within a computer environment with insufficient information. In applied research, one of the roads being explored is using speech synthesis and recognition in dialogue systems (Bilange, Button *et al.*, D. Luzzati), in which the human user is helped by the agent that is often only vocally represented, another involves agents or "\*-bots" for explaining normal use and workings of computer applications; the latter may also have a graphical representation. It seems clear that the tendency is to lift the artificial *it*'s up to the realm of the (human) *who*'s, at least in appearance, to help defend the new position of artefacts in these lesser-repetitive and tougher tasks.

And what if this does not work? What if this cannot properly lead to recreating interpersonal-like communication (*cf.* Schmidt 2004)?

Well it does for certain purposes. But the success of the scheduling, booking and other such information scenarios listed above is mainly due to the fact that the databases being scanned are done so with unvarying questions leading to fixed responses. In humanising the *it*'s that are designed to perform more personal tasks in traditional dialogical settings, the creators of agents seem to misjudge the problem of dealing with the *essences* of the different categories of beings involved — artificial and organic. The nature of the second type of task discussed here, those requiring the *social context* and rules of human communication in order to respond to the *inexact*, *underdetermined* or *unfinished* nature of the questions being asked would seem to indicate the creature explaining things for the user needs a more human aspect to make the dialogue work, when in fact, the totally opposite is true (*cf.* D. Luzzati 1989; Schmidt 2001). Are we really rendering the technological *it* service by placing it in a human-like corporal existence (two-dimensional or otherwise)? The answer to this question is most certainly "no" if the method being used is the wholly programmed-in one; it is too early to know for the '*learn-on-the-go*' option.

Within the framework of "agentising" all participants in a system to achieve explanatory simplification, this last question cannot go without its counterpart:

are we really rendering the natural human *who* service by placing him in a situation in which the associative connections of his interrogative thought patterns and the flirting 'here-and-now' of his dialogical capacities would be aborted by imposing limitations as reductive as pre-set questions? The answer to this one will always be no.

#### 4 The Dusk of Agents Without Faces

In agentism, the success/failure to shift focus from the nature of the entities carrying out actions to the actions themselves being performed is thus not only reliant upon the clarity-vagueness factor of tasks A-Life, AI and HCI Scientists wish to tackle when creating dialogue scenarios; it is also tributary to the very nature of the two types of entities —*who's* and *it's*— they are *trying* to absorb with the concept of agent. When we see two people interacting, say a teacher and a student co-constructing meaning in the course of their discourse about how a piece of software works, there seems to be something evasive to dialogue automation in the *immediateness of their relation-supported speech*. The nearly always unpredictable actions of the Self are obviously a function of those of the Other in communication, and *vice versa*. Likewise, it goes without saying that the mathematical strength of the machine cannot be duplicated by a mere human.

Perhaps we have simply come to the dawning of “de-agentisation”?

I do not know which notion should replace *agent* in modelling systems containing men and machines, or whether it can be replaced at all. But in tackling the problem of how to name, characterise, organise and *homogenise* various entities in such a system, scientists have built another problem, that of creating a new category that is *much* too vast to contain what they wish it to contain —*who's* and *it's*.

Do we not need a certain level of diversification *within* system description, specification, explanation and design to give the system adaptability? Though we may still be at the dawning typologies of A-Life creatures (*cf. the Proceedings of the RO-MAN Conference* in Berlin 2002), a typology of agents in society has been "complete" for centuries now: humans, other living things and objects). Quite simple at that; the these last 50 years has come to knock this off balance. So if our world of agents is becoming more complex, why should we simplify the terminology? Would not such an action be bound to fail?

De-agentising our approaches to modelling vast systems can contribute to helping *who*-users (humans) excel in environments including A-Life creatures on a large scale in the future. Re-humanising *who's* through detailed description capturing their singular aspects is necessary in order to design new situations and *types of communication* to create appropriate dialogue situations not only to integrate the human, but also to provoke a satisfactory knowledge acquisition process for other "agents" endowed with communication skills.

If we de-agentise now, what notion is this likely to bring into the crossroads of society, Computer Science and Robotics? Well, keeping in mind that my goal is to rebut agentism, answering such a question in a sufficient manner would mean defining what a non-agent is. De-agentising does not mean to take away the character of an entity that takes action (i.e. rendering it unable to act), but showing that it acts *either* by a

human-prescribed purpose (artificially) *or* through the ambition, will and desire of Man himself (the first step towards a typology of agents to include 'A-creatures').

De-homogenising the essences of the entities that can act within a system entails a certain will on the part of the scientist to understand and work with greater complexity than in a reductive "agentised system". This is necessary today in order to *properly* promote the varying potentials of each participant in a system. The notion of non-agent is meant to encompass the *degrees* of difference between the poles in the system with respect to their ability or non-ability to communicate. Communication—in the dialogical sense proposed by F. Jacques (1985)—is supported by the conventions of language use, a complex set of 'rules' inscribed in the societal existence of *who's*, but not *it's*. *It's* are perhaps making progress, but (entire) social recognition from humans outside the A-Life Circle is a long way off.

Humans cannot have the predictability factor of machines concerning their functions. The very *essence* of *who's* is to be *un-programmable*. *It's* cannot have full social status as long as they are produced and regulated by Man.

The notion of *non-agent* is meant to show in an extreme but logical way the *degrees* of differences (i.e. 0%) allowed between the various types of participants described in systems-oriented literature today. But it is the kind of thing that forces the scientist to modify his *Weltanschauung*<sup>6</sup>: as an outright negation of the notion that hinders the proper distribution of attributes to participants in the system—the notion of *agent*—, it lifts our focus from the *performing* of the action and re-centres our attention on the question of the *performer*, the one that *takes action*, the one that desires to bring about change to the world.

Trying to convince human society of the success of the human-machine relation is due to the *positivistic 'push'* in the Artificial Sciences or what some philosophers could—for the time being—qualify as over-enthusiasm...

The "generalisation" of the various participants in multi-agent systems by the concept of agent is no longer a valid action in the artificialisation of our society, a scientific and communitarian process which is now mature enough to manage the distinctions discussed here. People, computers and "creatures" produce acts which then belong to the past; in order to assign full meaning to an act, one must *know* the *who* or the *it* that is propagating the act. Further improved research in Artificial Life means being more exact in our use of terminology.

## 5 Pragmatic Considerations for Agency

Traditionally, the aim in Science has been based on precision. The argumentation I develop up until now points out that the use of the notion of *agent* in conversation and animation related fields goes against all sensible logic in Science to be precise. Using the notion of agent, as it is currently defined in the literature (i.e. under-defined), to refer to all participants in a system can lead one astray when it comes to organising these *varying capacities* as resources of the system. The loss of granularity in the description of entities hinders referring properly to what they are capable of doing. This is not compatible with modelling a system or showing its possible evolutions. It

---

<sup>6</sup> Worldview: the overall perspective from which one sees and interprets things.

is becoming increasingly necessary to reflect differences in complex systems comprised of humans and enticing objects like computers (i.e. in public services carrying educational, entertainment and cultural content).

This is all the more necessary when one submits unknown entities to the public eye, as such entities often force a person, however briefly, to *suspend key beliefs* about others and himself. In *Technik und Wissenschaft als Ideologie*, Habermas (1973) defines technology as intentionality *with a purposeful mission*. Based on this utilitarian view and the current usability of conversation-supporting technological "products", some may ask whether traditional Human-Computer Interface has not been high-jacked by mad scientists *playing Mother Nature*. Or are these scientists driven by an inner voice wrathfully telling them not to temper with Man's privileged position in the universe (i.e. "stop trying to recreate Man or deform his image!") to the point that they focalise on cartoon-like characters instead? Whatever the answer is to this question, most people spontaneously *understand* the "mission" of the socially animated in the classical sense: explaining how to use software (avoids heuristic searches), shifts focus and/or context (gets attention), conveying short messages concerning key information (saves time), etc.

If the main idea behind developing artificial creatures and characters and usage scenarios means accepting the diversities of physical manifestations *and minds*, would we not be better off to specify what we mean by "agent"? In general, it would seem that recent fields proposing technologies need rigorous presentation and argumentation to find a place within mainstream society. In any event, the specialist in animated pseudo-human agents is most certainly devising the necessary *forms* to transform the average person's *Weltanschauung* —his comprehensive conception of himself in the future world is that he is a *non-artificially* animated creature.

## 6 Categorisation and the Concept of Non-agent

Should we now consider the non-artificial —or rather the "natural" or the "organic"—creature to be the exception? Given the weight and number of pages written on artificial agents, we would have to. But this is only because there is a clean cut-off point between the literature about the artificial and that which could have been its inspiration, the human sciences, i.e. modal logic of individual persons in which the person carries out an action and is thus logically an actor or an agent. The difference is that being an agent in this latter field has nothing to do with the end result of the exercise —in computing the agent is the production. In the humanities, the agent is merely an ingredient to be used in a recipe, the resulting "cake" being a proposition on how the individual mind works in a societal context.

### 6.1 From the Motivation of the Concept to Its Meaning

Many authors have written on the topic of agents as largely expressed in the literature in computing (*cf.* the AAAI website). And on the fringe of such work, one would have to mention recent related research, such as that on Moral Agents proposed by

B.C. Stahl (2004<sup>7</sup>). *But what if restating or correcting the existing typologies of agents—or building new ones—is not the best action at this point?* I for one do not think that putting one's effort into this area intrinsically 'latched onto' the materialist agent could be considered as the result of a scientific process: after 50 some years, doing so has not lead to responding to the dialogical challenges (Turing 1950) launched for the machine no matter how we categorise the participants in the system. Quite likely, a more optimal action would be to *reposition* the existing typologies and their various versions with respect to the new context in which the natural agent is outnumbered (there is nothing ethical implied in this statement for the moment, I am reasoning purely about (future) situations on a quantificational basis). *Any* typologies concerning agents should undergo this treatment. The way to do this is to develop a "counter-concept", that was already eluded to above. That of a non-agent.

Might the reader be able to cite an example of a categorisation system specific to animated objects, conversational agents, social agents or any like creatures that makes explicit reference to a *non-agent* component? What would be the motive for it? Proposing such an anti-tradition entity for inclusion to the equation at this point is, first and foremost, to redirect the scientists attention to the fact that there are other points of view available. However, broadening horizons to the point of including 'reverse-thinking' concepts is the farthest one can go without totally changing subject areas. The fact that I am arguing for adopting this extreme position could be seen as the weak point in my discourse, but, may the reader be reassured, it is intentional. The term "non-agent" is synonymous with typical human being, truly dialogical language performer, an unsimulated human doer; to sum up, *any being that can mean*. It is the opposite of any machine or 'human behaviour copier' for the simple reason that a human being is not only composed of manifest entities (body and behaviour, or signs of these).

## 6.2 The Function of the Notion of Non-agent

I have devised the concept of non-agent for the immediate purposes of understanding the motivation to create animated characters, Artificial Life and related technical objects, as well as their explanatory downfalls. Its function is simple. I mean non-agent to be a beneficial element in designing language that can be understood by 'pro-agentistic' scientists. We must go to this extent because designers of the Artificial seem to be deaf to discourse on better differentiating human and machine-like agents (specialists in human-machine interaction generally try to merge them, as discussed above). Expressing similar ideas in the traditional way, within the Human Sciences (Psychology, Philosophy, Epistemology, Communication Sciences, etc.) and through their entry point journals, does not get the attention of those working in Computer Sciences, Computer Graphics and Animation, or any other related "rock-hard" (material-centred) scientific field for that matter. I do consider the virtual world to be physical as its goal is to produce a visible manifestation, even if the designer passes

---

<sup>7</sup> In wondering whether machines can be ascribed responsibility, Stahl speaks of computers that "produce statements about a human being or initiate actions that affect the moral rights and obligations of that human being", cf. STAHL B.C. (2004), "Information, Ethics and Computers", p. 68.



through a conceptual (i.e. non-physical) phase to this end. The break-off point between sciences interested in *how* and *why* actions are performed is quite evident in my mind—I cannot speak for others—; to my knowledge, they were never really fully able to get together on the matter and this is the primary driving force behind this article.

## 7 Conclusion

As I pointed out at the outset, humans are agents to a certain extent. In this context, “non-agent” would really mean a non-artificial agent. The difference I am trying to grasp and hold up to the light is the mind/body problem, *whether the body is organic or artificial*. So the key question brought about here is: *does or can machinery have mind?* This question is valid for all of the artistically animated creatures that are coming into being today. If we listen to their creators, they all aspire to human agency.

So what are the conditions for successful human agency? As D. Elgesem puts it, “[...] having an intention is not sufficient for agency while intentionality under some description is” (p. 4). Having mind depends on how ‘wide’ your definition of intentionality is; if it is outrageously large, you can call anything an agent. Otherwise tightening up the definition would logically exclude any artificial entities for they do not really have conscious control of their actions. Or do they? One might say that we cannot know. Perhaps AI creatures do have conscious feelings, intentions, beliefs etc., especially if we cannot prove the contrary. This would most certainly be right. But if we did tighten up the definition and were obliged to exclude some agents, which would we keep?

In test-bedding the terminology pertaining to agency, I deemed it was necessary to negate the notion of agent and analyse the consequences as well as to presuppose intentionality had an important role in disambiguating agent terminology. In comparing various agents, the default assignment value for Intentionality is not artificial beings but natural human ones.

## References

- The AAAI website, <http://www.aaai.org/AITopics/html/agents.html>
- Dascal, M.: Why does Language Matter to Artificial Intelligence? In: *Mind and Machines*, vol. 2, pp. 145–174. Kluwer Academic Publishers, Dordrecht, The Netherlands (1992)
- Davidson, D.: Agency. In: Binkley, R., et al. (eds.) *Agent, Action, and Reason*, University of Toronto Press, Toronto (1971)
- Dennett, D.: *The Intentional Stance*. Bradford/The MIT Press, Cambridge, MA (1987)
- Dreyfus, H.L.: *What Computers Still Can’t Do. A Critique of Artificial Reason*. The MIT Press, Cambridge, MA (1972)
- Elgesem, D.: The Modal Logic of Agency. *Nordic Journal of Philosophical Logic* 2(2) (December 1997)
- Habermas, J.: *La technique et la science comme ideologie*, Paris: Gallimard. [Technik und Wissenschaft als Ideologie] (1973)
- Hendler, J.: Is There an Intelligent Agent in Your Future? In: *Nature. Web Matters*, Macmillan Publishers, NYC (1999)
- Jacques, F.: *L’espace logique de l’interlocution*, Paris: Presses Universitaires de France (1985)

- Luzzati, D.: Recherches sur le dialogue homme-machine: modèles linguistiques et traitements automatiques, Thèse d'Etat, Université de la Sorbonne-Nouvelle, Paris III (1989)
- Poggi, I., Pélauchaud, C.: Performative faces. *Speech Communication* 26, 5–21 (1998)  
 NEC's web page: [http://www.incx.nec.co.jp/robot/PaPeRo/english/p\\_index.html](http://www.incx.nec.co.jp/robot/PaPeRo/english/p_index.html)
- Putnam, H.: *Language and Reality*, vol. 2. Cambridge University Press, Cambridge, MA (1975)
- Quine, W.V.O.: *Word and Object*. The MIT Press, Cambridge, MA (1960)
- Schmidt, C.T.A.: (forthcoming, with translation into Italian), *Having Difficulty with Identity*, *Teoria. Revista filosofia*, Edizioni ETS, Pisa
- Schmidt, C.T.A.: Of Robots and Believing. *Minds and Machines. Journal for Artificial Intelligence, Philosophy and Cognitive Science* 15(2), 195–205 (2005)
- Schmidt, C.T.: A Relational Stance in the Philosophy of Artificial Intelligence. In: *European Conference on Computing and Philosophy*, June 3-5, University of Pavia, Italy (2004)
- Schmidt, C.: Pragmatically Pristine, the Dialogical Cause. In: *Open Peer Community Invited Commentary, on Mele A., Real Self-deception, Behavioral and Brain Sciences*, vol. 20(1), Cambridge University Press, Cambridge, MA (1997)
- Searle, J.: *Speech Acts*. Cambridge University Press, Cambridge (1969)
- Shapiro, S., Rapaport, W.: Models and Minds: Knowledge Representation for Natural-Language Competence. In: Cummins, R., Pollock, J. (eds.) *Philosophy and AI: Essays at the Interface*, pp. 215–259. MIT Press, Cambridge, MA (1991)
- Smith, B.: (draft), *Ontology and Information Science*, *Stanford Encyclopaedia of Philosophy*
- Strawson, P.: On Referring, *Mind*, LIX(235) (1950)  
 Toshiba's web page, [http://www.toshiba.co.jp/about/press/2003\\_03/pr2001.htm](http://www.toshiba.co.jp/about/press/2003_03/pr2001.htm)
- Turing, A.: Computing Machinery and Intelligence, *Mind*, LIX (1950)
- Turkle, S.: *The Second Self, Computers and the Human Spirit*. Simon & Schuster, New York (1984)
- Vernant, D.: *Du discours à l'action. Etudes pragmatiques*, Paris: PUF (1997)
- Wittgenstein, L.: *Philosophical Investigations*, Oxford, Blackwell (1953)
- Zlatev, J.: The Epigenesis of Meaning in Human Beings, and Possibly in Robots. In: *Minds and Machines*, n 11, pp. 155–195. Kluwer, Dordrecht (2001)

# Inductive Logic Programming Algorithm for Estimating Quality of Partial Plans

Sławomir Nowaczyk and Jacek Malec

Department of Computer Science  
Lund University, Sweden

slawek@cs.lth.se, jacek@cs.lth.se

**Abstract.** We study agents situated in partially observable environments, who do not have the resources to create conformant plans. Instead, they create conditional plans which are partial, and learn from experience to choose the best of them for execution. Our agent employs an incomplete symbolic deduction system based on Active Logic and Situation Calculus for reasoning about actions and their consequences. An Inductive Logic Programming algorithm generalises observations and deduced knowledge in order to choose the best plan for execution.

We show results of using PROGOL learning algorithm to distinguish “bad” plans, and we present three modifications which make the algorithm fit this class of problems better. Specifically, we limit the search space by fixing semantics of conditional branches within plans, we guide the search by specifying relative relevance of portions of knowledge base, and we integrate learning algorithm into the agent architecture by allowing it to directly access the agent’s knowledge encoded in Active Logic. We report on experiments which show that those extensions lead to significantly better learning results.

## 1 Introduction

Rational, autonomous agents able to survive and achieve their goals in dynamic, only partially observable environments are the ultimate dream of AI research since its beginning. Quite a lot has already been done towards achieving it, but dynamic environments still remain a big challenge for autonomous systems.

One of the major ways of coping with uncertainty and lack of knowledge about current situation is to exploit previous experience. In our research we are interested in developing rational, situated agents that are aware of their own limitations and can take them into account. Due to limited resources and the necessity to stay responsive in a dynamic world, situated agents cannot be expected to create complete plans for achieving their goals. They need to consciously alternate between reasoning, acting and observing their environment, or even do all those things in parallel. We aim to achieve this by making the agent create short partial plans and execute them, learning more about its surroundings throughout the process.

The agent creates several partial plans and reasons about usefulness of each one, including what knowledge can it provide. It generalises its past experience to evaluate how likely particular a plan is to lead to agent’s goal. The plans are conditional (i.e. actions to be taken depend on observations made during execution), which makes them

more generic and means their quality can be estimated more meaningfully. In addition, the agent will judge whether it is more beneficial to begin executing one of those plans immediately or rather to continue deliberation.

In general, the ultimate goal of this architecture is to make it possible to put together state-of-the-art solutions from several different areas of Artificial Intelligence. Despite multiple attempts, both ones done in the past and those still in progress, the vast majority of AI research is being done in specialised subfields and it is our belief that neither of these subfields *alone* can give us truly intelligent, rational agents. Our architecture, which to the best of our knowledge is novel, may be one way to integrate such approaches.

The goal of this paper is to show how a well-known learning algorithm can be modified to better fit one class of problems faced by rational agents. Sections 2 to 6 describe the architecture of our agents, in order to provide background for our work and to justify both the usefulness of our modifications and their applicability. It is followed by initial experimental results using vanilla PROGOL in section 7 and discussion where and how can they be improved upon in section 8. We claim that the new algorithm is better suited for rational agents, while still fully agnostic with regard to the actual domains in which those agents operate.

## 2 Agent Architecture

The architecture of our agent consists of four main functional modules. Each of them is responsible for a different part of agent’s rationality, but the overall intelligence is only achievable by the interactions of them all.

The *Deductor* module is the one responsible for classical “reasoning”. It uses a logical formalism based on combination of Active Logic and Situation Calculus (as introduced in [16]) in order to find out consequences of the agent’s current beliefs. Based on the domain knowledge and previous observations, it analyses possible actions and predicts what will be the effects of their execution.

The second module is *Planner*, which generates partial, conditional plans applicable in the agent’s current situation. The third main module, *Actor*, oversees Deductor’s reasoning process and evaluates plans the latter has come up with, trying to find out which is the most useful one to execute.

Finally, the *Learner* module analyses the agent’s past experience and induces rules for estimating quality of plans. Results of learning process are used both by Deductor and by Actor. In particular, since the plans Deductor reasons about are partial (i.e. they do not — most of the time — lead all the way to the goal) it can be very difficult to predict whether a particular plan is a step in the right direction or not. Using machine learning techniques is one possible way of achieving this.

## 3 Experimental Domains

Throughout this paper we use a simple game called Wumpus, a well-known testbed for intelligent agents [23], to better illustrate our ideas. The game is very simple, easy to understand, and humans can play it effectively as soon as they learn the rules. For

artificial agents, however, this game — and other similar applications, including many of practical importance — remain a serious challenge, mostly due to the combinatorial explosion of game states that need to be considered. It is impossible to analyse all of them explicitly, and while symbolic reasoning shows promise to provide effective generalisations over them, at the current state of the art one still needs to test ideas and prototypes on small, artificial examples.

The Wumpus game is played on a square board. There are two characters, the player and the Wumpus. The player can, in each turn, move to any neighbouring square, while the Wumpus does not move at all. Position of the monster is not known to the player, he only knows that it hides somewhere on the board. Luckily, Wumpus is a smelly beast, so whenever the player enters some square, he can immediately notice if the creature is in the vicinity. The goal of the game is to find out the exact location of the monster, by moving throughout the board and observing on which squares does it smell. At the same time, if the player enters the square occupied by the beast, he gets eaten.

We also use a second domain, a modified version of “king and rook vs king and knight” chess ending. Since we are interested in partially unknown environments we assume, for the sake of experimentations, that the agent does not know how the opponent’s king is allowed to move — *a priori*, any move is legal. The agent will need to use learning to discover what kinds of moves are actually possible.

## 4 Deductor

Deductor performs logical inference and directly reasons about the agent’s knowledge. In particular, it is the module which analyses both current state of the world and how it will change as a result of performing a particular action. To this end, the agent uses a variant of Active Logic [20], augmented with some ideas from Situation Calculus [21].

Active Logic is a reasoning formalism which, unlike classical logic, is concerned with the *process* of performing inferences, not just the final extension of the entailment relation. In particular, instead of the classical notion of theoremhood, AL has *i-theorems*, i.e. formulae which can be proven *in i steps*. This allows an agent to reason about *difficulty* of proving something, to retract knowledge found inappropriate and to resolve contradictions in a meaningful way. An in-depth description of Active Logic can be found in [20].

Following the ideas of [16], we have decided to augment AL with concepts from Situation Calculus. Since the agent needs to reason about effects of executing plans in various situations, we index formulae both with the current situation and with the plan being considered. Therefore, a typical formula can be:  $Knows(s, p, Neighbour(a2, b2))$ , meaning “the agent knows that after executing plan  $p$  in situation  $s$ , squares  $a2$  and  $b2$  will be adjacent.”

From agent’s point of view, the most interesting formulae are ones of the form:  $Knows(s, p, Wumpus(b3)) \vee Knows(s, p, Wumpus(c2))$ , meaning “an agent knows that after executing plan  $p$  in situation  $s$ , it will *either* know that there is Wumpus on  $b3$  or that there is Wumpus on  $c2$ ”. Which of the “or” clauses will actually be true depends on the observations made while acting. This is exactly the kind of knowledge that we are interested in agent inferring — it *does* tell important things about quality of the plan

being considered. For a human “expert,” such  $p$  looks like a good plan. The goal of our research is to make *an agent* be able to reason about plans in exactly this way.

In order to make plan evaluation more meaningful, we allow plans to not only be simple (sequential) but also *conditional*, i.e. to contain branches where actions depend on observations made during acting. We believe that such conditional plans will be, in many domains, much easier to classify as either good or bad ones, since they contain more *generic* knowledge and their applicability is greatly expanded.

## 5 Actor and Planner

The Actor module is an overseer of Deductor and works as a controller of the agent as a whole. In its ultimate form, it is expected to do three main things. First, it guides the reasoning process by making it focus on the plans most likely to be useful. Second, it decides when enough time has been spent on deliberation and no further interesting results are likely to be obtained. Third, it makes decisions which plan, from Deductor’s repertoire, to execute.

In this paper we have decided to focus more on the interactions between learning and deduction, so both Planner and Actor have been significantly simplified. Planner does not use any heuristics and simply creates all possible plans, although existing planners can be used to efficiently create only the “reasonable” plans. Also, Deductor uses an incomplete reasoner which always terminates, therefore Actor does not need to decide *when* to begin plan execution — it lets Deductor infer everything it can about each of the available plans and chooses the best one based on all available information.

## 6 Learner

The ultimate goal of the learning module is to provide Actor with knowledge necessary to choose the best plan for execution and to stop deliberation when too much time has been spent on it without any new interesting insights.

A step in this direction is to learn how to detect “bad” plans early, so that Deductor does not waste time deliberating about them. In our experimental domains we have defined bad plans to be those which can kill the agent (for Wumpus), and those that lead to losing the rook (for Chess).

One question is how to represent situations and plans in the way most suitable for learning. We have decided to encode plan and its branches as additional arguments to the domain predicates. The first step of a plan is an unconditional one — the agent simply decides how to move in a given situation. For Wumpus domain, the rest of the plan consists of two branches (called *left* and *right*, with *left* being taken iff it smells on the newly-visited square). For Chess, there are three explicit branches (each specifying expected move of the opponent and the agent’s response, without any meaning assigned to their order) and, additionally, a *default* branch, which will be executed whenever the opponent makes any move other than those three. It is our belief that such representation is sufficiently general to be usable across many different domains. For example, a predicate  $Position(p1, left, a2)$  means that after the *left* branch of plan  $p1$  is executed, the agent will be on square  $a2$ .

In the experiments reported here, we assume that the agent has perfect knowledge about which plans (training examples) are bad ones. This is a justified assumption for Chess domain, where the opponent does not make trivial mistakes and whenever it is possible for him to capture the rook, he will do so. In Wumpus, the distinction is not so clear — it is possible that the agent will get lucky and not die even though it executes a dangerous plan, simply because the beast is in an favourable position.

## 7 Results of Initial Experiments

We have used an ILP algorithm PROGOL [15] for our experiments, since it is among the best known ones and its author has provided a fully-functional, publicly available implementation. In a previous paper ([17]) we have presented results of learning to distinguish bad plans early. We have shown that PROGOL is able to find the correct hypothesis from as few as 30 randomly-chosen examples. Such a hypothesis allows the agent to save up to 70% of its reasoning time, since it does not need to waste resources analysing plans which are known to be useless. Those results, however, required providing additional domain knowledge specifically for the purpose of learning.

The results we have obtained can be seen in figure 1. The lowest curve corresponds to an experiment where we used as little domain-specific knowledge as possible, in particular we have not even provided PROGOL *mode declarations*. For the second curve, marked “Full Knowledge Base,” we have provided exactly them. Clearly, even such small amount of additional knowledge greatly improves the quality of learning. PROGOL still failed to learn perfectly, though, and we have identified that it was due to the excessive amount of input knowledge. Therefore, for our third experiment, depicted in figure 1 as curve marked “Relevant Predicates Only,” we have selected the most relevant parts of knowledge generated by Deductor and presented only them to PROGOL, finally achieving satisfactory results.

It is interesting to note that as few as five *hand-chosen* example plans suffice for PROGOL to learn the correct definition for the Wumpus domain, which opens up interesting possibilities for an agent to *select* learning examples in a clever way.

## 8 Algorithm Modifications

PROGOL is based on the idea of *inverse entailment* and it employs a covering approach similar to the one used by FOIL [13], in order to generate hypothesis consisting of a set of clauses which cover all positive examples and do not cover any negative ones. It starts with positive example  $e$  and knowledge base  $B$  and creates a set  $\perp$  which is the set of all ground literals true in all models of  $B \wedge \bar{e}$ . Due to the properties of inverse entailment, the complete set of hypotheses  $\mathcal{H}$  consists of all the clauses which  $\Theta$ -subsume *sub-saturants* of  $\perp$ . PROGOL uses mode declarations to limit the size of  $\perp$  and refinement operator  $\rho$  to efficiently search only a subset of  $\mathcal{H}$ . More details can be found in [15].

We have made three modifications to the PROGOL algorithm in order to make it better suited for rational agents. All of them were inspired by the difficulties PROGOL

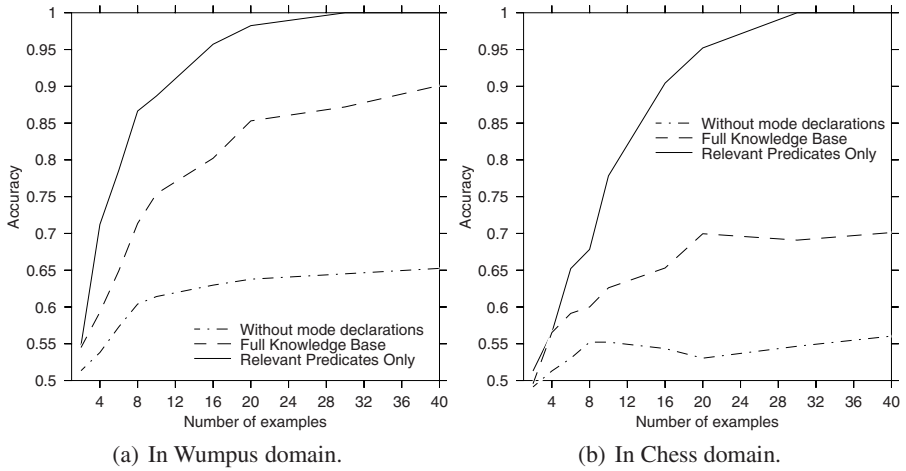


Fig. 1. Results of learning

encountered when solving our problem, difficulties which apparently resulted from mismatches between its assumptions and the properties of problem it faced.

### 8.1 Branch Awareness

First of all, it is well known that PROGOL is sensitive to the arity  $k$  of predicates in its background knowledge, since the cardinality of sub-saturant set is bounded by  $n^k$ . Often, this is not an issue, since the arity of predicates is typically kept low. In our setting, however, many predicates contain two additional arguments, namely the plan and its branch. Instead of simply saying  $Wumpus(a1)$  to state monster’s position, we need to say  $Wumpus(p_1, b_1, a1)$ , i.e. in branch  $b_1$  of plan  $p_1$ , Wumpus is on  $a1$ .

This has lead to a problem where PROGOL was unable to learn, in the Chess domain, the correct hypothesis in the form we originally wanted. We have specified a predicate *chooseBranch* with mode declaration “modeb(4, chooseBranch(#step,-step))?”, which would be useful for “naming” a particular branch, i.e. binding it to a variable for following predicates to use. This way we could guarantee that *the same* branch is used throughout the whole clause.

In particular, we intended one of the clauses in the hypothesis to be

$$badPlan(A) : -chooseBranch(default, B), notProtected(A, B, rook), distanceTwo(A, B, rook, black-king).$$

meaning “whenever in the default branch the opposing king is close to our rook, the plan is dangerous.”<sup>1</sup> Such a hypothesis, however, proved to be too difficult for PROGOL and

<sup>1</sup> Observe that in the default branch we do not know the exact move an opponent has made, therefore  $Position(black-king)$  is the “old” position and we need to be extra careful about safety of our rook.



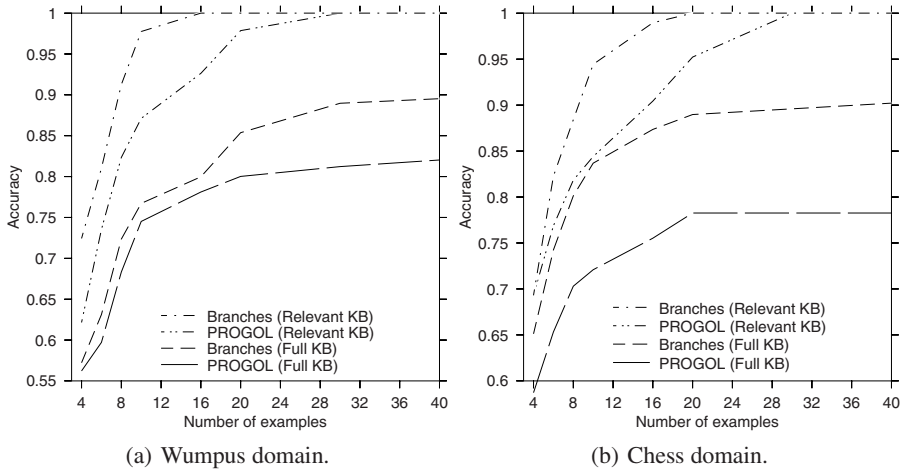


Fig. 2. Branch awareness

despite numerous attempts, we have not managed to “convince” it to learn it. We had to settle for a similar, but somewhat less natural, definition of

$$\begin{aligned}
 \text{badPlan}(A) &: \text{--notProtected}(A, \text{default}, \text{rook}), \\
 &\text{distanceTwo}(A, \text{default}, \text{rook}, \text{black-king}).
 \end{aligned}$$

Notice that this one uses separate, unrelated constants as arguments of *notProtected* and *distanceTwo* predicates. In our experiments PROGOL required some additional examples before it learned not to use two *different* branch there. For example, a clause

$$\begin{aligned}
 \text{badPlan}(A) &: \text{--notProtected}(A, b_1, \text{rook}), \\
 &\text{distanceTwo}(A, b_2, \text{rook}, \text{black-king}).
 \end{aligned}$$

is, obviously, not good — whether the rook is protected in some branch  $b_1$  has no meaningful relation to the distance between rook and black king in  $b_2$ .

To fix this issue, we have modified the PROGOL learning algorithm to *transparently* hide branches, i.e. to automatically restrict knowledge base to only contain facts from a single branch when reasoning. This has lowered the arity of most predicates by one and allowed for better hypothesis to be found from fewer examples.

It is questionable, however, whether this is not too drastic a measure, since one can imagine domains where different plan branches are not completely independent and it would be beneficial to reference two or more of them from a single hypothesis clause. If such a need arises, it can be satisfied by extending the mode declaration language.

Figures 2(a) and 2(b) compare the results of learning using vanilla PROGOL against one which contains built-in knowledge about plans and branches. It can be easily seen that such a modification leads to better learning results.

It is also important to note that the hypothesis learned by the modified algorithm is of better quality, since it contains — as explained above — a variable bound to the default branch, instead of having duplicated constants.

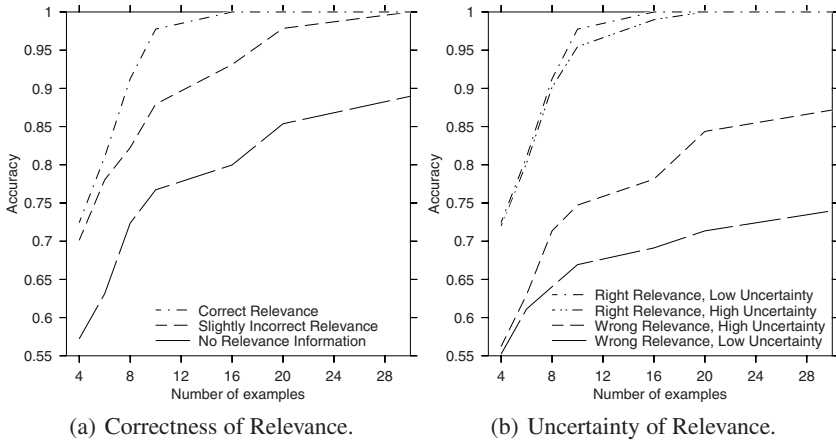


Fig. 3. Relevance experiments in Wumpus domain

## 8.2 Knowledge Relevance

As can be seen in results mentioned in section 7 too much knowledge can significantly decrease the performance of the learning algorithm. In a typical ILP setting it is not of crucial importance to quantify which parts of knowledge are more and which are less relevant [1] — since an expert provides the complete knowledge in the first place, if she knows that some parts are irrelevant, she will simply omit them.

The situation is somewhat different in the agent setting, since the knowledge from which we are learning does not come *directly* from an expert, but rather from Deductor. It is a product of initial domain description, the observations the agent has made, and its own deductive process — and all those elements interact in complex ways.

The amount of irrelevant knowledge in such a conglomerate is, typically, rather high. We have shown that choosing the relevant parts only can lead to much better learning results. In an earlier paper [18] we have analysed how our setting can be seen as a way to estimate relevance of pieces of knowledge, and how automatic heuristic procedure can be constructed to select which knowledge is most useful.

It is important, however, to stress that any kind of automatic procedure to estimate the relevance of data is very approximate in nature and even though it works very well on our domains, we do not want to *over-commit* to its results. In particular, the idea to completely remove the predicates deemed irrelevant from the *whole* learning process is very dangerous. The price of making a mistake seems to be too high.

What we want, instead, is a way for the learning algorithm itself to take into account the relative relevance of knowledge when it is building the hypothesis. In the case of PROGOL, the perfect place to take knowledge relevance into account is in the  $\rho$  operator, where current hypothesis candidate is being generalised by new literals.

In Figure 3 we report our analysis of usefulness of building relevance considerations into the algorithm itself. Part (a) shows that even providing relevance information

<sup>2</sup> Although in the field of machine learning there is a lot of interesting work being done on this topic, especially with regard to data mining. Good surveys are, for example, [4] and [26].

which is not entirely correct can be beneficial. Curve “Correct Relevance” depicts unambiguously correct information about relevant predicates and corresponds exactly to “Branches (Relevant KB)” from figure 1; curve “No Relevance Information” depicts complete lack of relevance information and corresponds to “Branches (Full KB)”.

Part (b) presents how uncertainty of relevance information influences the results (low uncertainty is the case when relevant predicates have estimate around 1 and irrelevant have estimate around 0, while high uncertainty is the case where relevant predicates have estimate slightly above 0.5 and irrelevant ones have estimate slightly below 0.5). An interesting fact is that increasing uncertainty of correct relevance information lowers quality of learning, but not significantly. On the other hand, as it can be expected, providing incorrect information with high certainty significantly disrupts the learning.

### 8.3 Using Active Logic

We have also added the ability to directly access agent’s knowledge, expressed in Active Logic, without an additional step of data transformation. Such direct access includes the transformation from *open world* semantics used by the Deductor to the *closed world* semantics used by PROGOL.

## 9 Related Work

Combination of planning and learning is an area of active research, in addition to the extensive amount of work being done separately in those respective fields.

The first to mention is [9], which presented results establishing conceptual similarities between explanation-based learning and reinforcement learning. In particular, they discussed how Explanation-Based Learning can be used to learn action strategies and provided important theoretical results concerning its applicability to this aim.

There has been significant amount of work done in learning about what actions to take in a particular situation. One notable example is [10], where author showed important theoretical results about PAC-learnability of action strategies in various models. In [14] author discussed a more practical approach to learning Event Calculus programs using Theory Completion. He used extraction-case abduction and the ALECTO system to simultaneously learn two mutually related predicates (*Initiates* and *Terminates*) from positive-only observations. Recently, [11] developed a system which learns low-level actions and plans from goal hierarchies and action examples provided by experts, within the SOAR architecture. Yet another fresh work close to this approach is documented in [12], where *teleoreactive logic programs*, possibly even recursive ones, are used for representing the action part of an agent. On top of it a learning mechanism, quite similar to ILP, is employed for improving the existing action programs.

In the general field of Inductive Logic Programming, there is a large number of systems being developed, such as CLAUDIEN [8], MOBAL [24], Charade [22], Rulex [1] and others. We have based our work on PROGOL mainly because it is the most popular one and various researchers have been working on improving multiple aspects of it, for example [27] and [2]. To the best of our knowledge, though, nobody has tried yet to learn to classify conditional partial plans.

Yet another track of research focuses on (deductive) planning, taking into account incompleteness of agent's knowledge and uncertainty about the world. Conditional plans, generalised policies, conformant plans, universal plans are the terms used by various researchers [6,19,25,3] to denote in principle the same idea: generating a plan which is "prepared" for all possible reactions of the environment. This approach has much in common with control theory, as observed in [5] or earlier in [7]. We are not aware of any such research that would attempt to integrate learning into this approach.

## 10 Conclusions

We are developing an architecture for rational agents that combine planning, deductive reasoning, inductive learning and time-awareness in order to operate successfully in dynamic environments. Our agent creates conditional partial plans, reasons about their consequences using an extension of Active Logic with Situation Calculus features, and employs ILP learning to generalise past experience in order to distinguish good plans from bad ones.

In this paper we report on our experiments with using PROGOL learning algorithm to identify bad plans early, in order to save agent the (wasteful) effort of deliberating about them. We show that it is possible to adapt a general purpose learning tool to better fit the specific requirements of learning to classify plans, and that such adaptation significantly improves the results obtained. We also show that ILP approach works fine within the framework of the architecture we are developing.

The research presented here can be continued in many different directions. The first thing we need to focus our attention on is the way to handle uncertainty about training data — again, there is a lot of work being done in this area, but there are some rather unique properties of training data in the rational agent case and we feel that it is important to take advantage of them to the highest degree possible.

Second step is to, instead of creating a classifier, devise an algorithm to efficiently "learn to compare", in the sense that the real goal of the agent is to choose *the best* plan available. Discarding bad plans is a step in this direction, but the classification approach is not necessarily the right one when the "least bad among dangerous" or the "most rewarding among marvelous" is to be selected.

In addition, for rational agents there is the very interesting notion of "experiment generation", since they often do not learn for a pre-prepared set of training examples, but rather face the complex *exploration vs exploitation* dilemma. How to act in a way which both provides short term rewards (or, at the very least, keeps the agent safe) and at the same time offers a chance to learn something new is often far from obvious.

Finally, the architecture we are presenting here is still evolving and the functionality of every module will be expanded in the future.

## References

1. Andrews, R., Geva, S.: Rule extraction from local cluster neural nets. *Neurocomputing* 47 (2002)
2. Badea, L., Stanciu, M.: Refinement operators can be (weakly) perfect. In: Džeroski, S., Flach, P.A. (eds.) *Inductive Logic Programming*. LNCS (LNAI), vol. 1634, Springer, Heidelberg (1999)

3. Bertoli, P., Cimatti, A., Traverso, P.: Interleaving execution and planning for nondeterministic, partially observable domains. In: European Conference on Artificial Intelligence, pp. 657–661 (2004)
4. Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. *Artificial Intelligence* 97(1-2), 245–271 (1997)
5. Bonet, B., Geffner, H.: Planning and control in artificial intelligence: A unifying perspective. *Applied Intelligence* 14(3), 237–252 (2001)
6. Cimatti, A., Roveri, M., Bertoli, P.: Conformant planning via symbolic model checking and heuristic search. *Artificial Intelligence* 159(1-2), 127–206 (2004)
7. Dean, T., Wellman, M.P.: *Planning and Control*. Morgan Kaufmann, San Francisco (1991)
8. Dehaspe, L., De Raedt, L., Laer, W.: *Claudien: The clausal discovery engine user's guide, The CLAUsal Discovery ENgine User's Guide*, Katholieke Universiteit Leuven (1996)
9. Dietterich, T.G., Flann, N.S.: Explanation-based learning and reinforcement learning: A unified view. In: *Int. Conf. on Machine Learning*, pp. 176–184 (1995)
10. Khardon, R.: Learning to take actions. *Machine Learning* 35(1), 57–90 (1999)
11. Könik, T., Laird, J.E.: Learning goal hierarchies from structured observations and expert annotations. *Machine Learning* 64, 263–287 (2006)
12. Langley, P., Choi, D.: Learning recursive control programs from problem solving. *Journal of Machine Learning Research* 7, 493–518 (2006)
13. Mitchell, T.M.: *Machine Learning*. McGraw-Hill Higher Education, New York (1997)
14. Moyle, S.: Using theory completion to learn a robot navigation control program. In: Matwin, S., Sammut, C. (eds.) *ILP 2002. LNCS (LNAI)*, vol. 2583, Springer, Heidelberg (2003)
15. Muggleton, S.: Inverse entailment and Progol. *New Generation Computing*, Special issue on Inductive Logic Programming 13(3-4), 245–286 (1995)
16. Nowaczyk, S.: Partial planning for situated agents based on active logic. In: *ESSLLI 2006. Workshop on Logics for Resource Bounded Agents* (2006)
17. Nowaczyk, S., Malec, J.: Learning to evaluate conditional partial plans. In: *Sixth International Conference on Machine Learning and Applications* (December 2007)
18. Nowaczyk, S., Malec, J.: Relative relevance of subsets of agent's knowledge. In: *Workshop on Logics for Resource Bounded Agents* (September 2007)
19. Petrick, R.P.A., Bacchus, F.: Extending the knowledge-based approach to planning with incomplete information and sensing. In: *International Conference on Automated Planning and Scheduling*, pp. 2–11 (2004)
20. Purang, K., Purushothaman, D., Traum, D., Andersen, C., Perlis, D.: Practical reasoning and plan execution with active logic. In: Bell, J. (ed.) *Proceedings of the IJCAI-99 Workshop on Practical Reasoning and Rationality*, pp. 30–38 (1999)
21. Reiter, R.: *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press, Cambridge (2001)
22. Ricci, F., Mam, S., Marti, P., Normand, V., Olmo, P.: CHARADE: a platform for emergencies management systems. Technical Report 9404-07, Povo (1994)
23. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 2nd edn. AI. Prentice-Hall, Englewood Cliffs (2003)
24. Sommer, E., Emde, W., Kietz, J.-U., Wrobel, S.: *Mobal 3.0 user guide*
25. van der Hoek, W., Wooldridge, M.: Tractable multiagent planning for epistemic goals. In: *First International Conference on Autonomous Agents and Multiagent Systems* (2002)
26. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. *Artificial Intelligence Review* 18(2), 77–95 (2002)
27. Yamamoto, A.: Improving theories for inductive logic programming systems with ground reduced programs. Technical report, AIDA9619, Technische Hochschule Darmstadt (1996)

# Modeling Emotion-Influenced Social Behavior for Intelligent Virtual Agents

Jackeline Spinola de Freitas<sup>1,2</sup>, Ricardo Imbert<sup>2</sup>, and João Queiroz<sup>1,3</sup>

<sup>1</sup> School of Electrical and Computer Engineering  
State University of Campinas P.O. Box 6101 – 13083-852 Campinas-SP, Brazil

<sup>2</sup> Computer Science School, Universidad Politécnica de Madrid,  
Campus de Montegancedo, s/n, 28660 Boadilla del Monte, Madrid, Spain

<sup>3</sup> Research Group on History, Philosophy, and Teaching of Biological Sciences, Institute of  
Biology, Federal University of Bahia, Salvador-BA, Brazil

{jspinola, queirozj}@dca.fee.unicamp.br, rimberty@fi.upm.es

**Abstract.** In the last decades, cognitive and neuroscience findings about emotion have motivated the design of emotional-based architectures to model individuals' behavior. Currently, we are working with a cognitive, multi-layered architecture for Agents, which provides them with emotion-influenced behavior and has been extended to model social interactions. This paper shows this architecture, focusing on its social features and how it could be used to model emotion-based agents' social behavior. A model of a prey-predator simulation is presented as a test-bed of the architecture social layer.

**Keywords:** Emotion-Based Architecture, Intelligent Virtual Agents, Social Interaction, Computational Simulation.

## 1 Introduction

Brand-new findings in neuroscience concerning the mechanisms, functions, and nature of emotions have promoted a review of the association between emotion, reason and logical behavior in human beings [1], [2], [3], [4]. Based on the evidence that emotions play a significant role in diverse cognitive processes, and that they are essential for problem solving and decision making, [1], [5], [6], [7], [8], Computer Science and Artificial Intelligence have started to model perception, learning, decision processes, memory, and other functions through emotion-based agents.

One particular application of these studies is the design of emotion-based architectures to model individuals' behavior. We have been working with a cognitive, multi-layered architecture for agents, called COGNITIVA, which provides them with emotion-influenced behavior. It includes a flexible model that allows setting up dependencies and influences among elements such as *personality traits*, *attitudes*, *physical states*, *concerns* and *emotions* on agents' behavior.

The architecture is generic, but provides mechanisms that allow adapting it to specific contexts through a progressive specification. COGNITIVA has been used to model agents, and more particularly, Intelligent Virtual Agents (IVAs) in contexts of

very different nature, such as virtual auctions bidders agents in Vickrey Auctions [9], virtual zebras and lions in a 3D virtual African savannah [9], [10] and to model virtual storytelling characters [11]. The resulting behavior of emotion-influenced IVAs in such experiments was proper and coherent, showing that the inclusion of emotion in such systems does not entail loss of control.

Now, COGNITIVA is been tested in social contexts, in which emotions have been considered of paramount importance. For instance, [12] argue that recent advances in the understanding of the intrapersonal characteristics of emotions have facilitated the complementary study of the interpersonal functions of emotions and research has begun to address the consequences of emotion beyond the individual, focusing on the ways emotions are embedded within ongoing interactions [13], [14], [15], [16], [17]. [14] and [16] state that emotions are dynamic, relational processes that coordinate the actions of individuals in ways that guide their interactions toward more preferred conditions and thus organize behavioral and cognitive response within the individual as well as interactions between individuals [14], [18], [19].

With such background knowledge, we have been working to include the appropriate psychological concepts also at the social layer of COGNITIVA, to model IVAs emotion-based behavior in social contexts, improving their believability and, even, their performance. The number of emotion-based architectures that contemplates social interaction has increased in the last two decades. However, differently from COGNITIVA, most of them focus mainly on user-agent interaction, like chatter bots, virtual storytelling characters and life-like characters for games and user interfaces. Additionally, most of them are context-dependent and do not allow their validation as a generic architecture ([20], [21]) or try to imitate unknown brain processes ([22]).

In order to discuss our research, in the following section, COGNITIVA and its main components are succinctly described (details in [10], [11]). The social layer of this architecture is totally connected with all of these components, but, since it is new, we have opted for presenting most of its characteristics separately, in section 3. Section 4 describes the social simulation context that serves as test-bed for the new features of the architecture. Finally, in section 5 we present some final comments.

## 2 A Review of COGNITIVA

As we have mentioned, COGNITIVA is a multilayered agent-oriented architecture and covers several kinds of behavior: reactive, deliberative and social. Each behavior comes from a related architecture layer, such that:

- The *reactive layer* is responsible for providing the agent with immediate responses to changes in its environment;
- The *deliberative layer* provides the agent with goal-directed behavior, from the individual perspective and abilities the agent has *per se*;
- The *social layer* also provides the agent with goal-directed behavior, but also considering the existence of other agents, the interaction with them and the potential use of their abilities to achieve personal and/or global goals.

The cognitive architecture receives inputs through an *Interpreter*, which filters and transforms the perceptions coming from the sensors into understandable units (*percepts*) that can be sent to the rest of the cognitive processes.

IVA's internal representation of every information source is called *Beliefs*, including knowledge about the environment (places and objects), about other IVA (individuals), and, even, about the IVA itself. For their management, COGNITIVA structures them into a taxonomy, structured as follows:

- *Beliefs* related to *Defining Characteristics* (DC), which describe the general traits of places, objects and individuals, and are important to identify them;
- *Beliefs* related to *Transitory States* (TS), characteristics whose values represent the current state of places, objects and individuals;
- *Beliefs* related to *Attitudes*, parameters that determine the behavior of an IVA towards other environmental components (places, objects and individuals).

A subset of IVA's *Beliefs* is related to the IVA itself and is fundamental to the architecture workings. This subset constitutes what is called the IVA's *Personal Model*, and contains representation of *personality traits*, *moods*, *physical states*, *attitudes* and *concerns*. These components are organized as: (i) DC such as *personality traits*, whose values determine coherent and stable behavior of the IVA; (ii) TS such as *moods* and *physical states*, identifying respectively the current state of the *mind* and the *body* of the IVA; and (iii), its *attitudes* towards others. COGNITIVA also proposes the use of *concerns* to represent the range of desirable values of the IVA's *Transitory States* at a specific moment, particularly, for its *emotions* and *physical states*. Each *concern* has an associated priority and an upper and a lower acceptable thresholds. Deliberative and social processes also manage the *concerns* threshold values to control reactive actuation whenever it is needed [10].

Many of these elements are intertwined. The relationship among them reveals the influence that each one of them has on the others as follows: (i) *Personality traits* exert an important influence on *moods*. For instance, before the same dangerous situation, a courageous IVA might feel less *fear* than a pusillanimous one; (ii) The set of the IVA's *Attitudes* has some influence on the *moods* it experiences. For example, the presence of a predator in an environment will increase prey *fear*, because of its attitude of apprehension towards predators; (iii) *Personality traits* influence *Attitudes*. E.g.: an altruist character may decide to interrupt its activities to serve another agent; (iv) *Physical states* influence *moods*. E.g., a hungry virtual animal might be more susceptible to irritation; (v) *Personality traits* exert some influence on *concerns*, and specify them according to individual characteristics. E.g., a courageous animal might have different upper and lower thresholds of risky avoidance than a coward one.

The set of these special *Beliefs* represents the state of an IVA, and has a strong influence on its behavior. That is why we say that emotions are not considered just as a component that provides the system with some 'emotional' attribute; *beliefs* have been designed to influence decisively on the IVA's behavior. During agent's life cycle, an IVA must be capable of maintaining information about what has happened in previous states. It maintains its *Past History* that, along with perceptions and *beliefs*, is used to improve the IVA's behavior selection. To update *beliefs* and *past history*, COGNITIVA incorporates the concept of *expectations*, which captures the IVA's predisposition towards an event that has happened or can happen. The expectations about an event are valued in terms of its probability of occurrence (*Expectancy*) and the desirability of its occurrence (*Desire*). Confirmation or disconfirmation of the occurrence of a given event generates a rich set of emotions.



The reactive layer responds immediately to events produced by changes in the environment through *Reflex Processing* and *Conscious Reaction Processing*. The deliberative layer explores the (autonomous) abilities an IVA has in order to achieve its objectives through goal-oriented behavior.

Both the deliberative and the social layers base their operation on two central concepts: *goals* and *plans*. *Goals* represent the objectives the IVA intends to achieve and, thus, direct its behavior. *Goals* are characterized by (i) an objective situation pursued; (ii) the current state of the goal (planned, achieved, cancelled...); (iii) their importance; (iv) a time stamp (to check goal validity); and (v) an expiry time. *Goals* can be produced from two different perspectives: according strictly to the individual abilities of the IVA (*Deliberative Goals Generator*), or considering interactions with other IVAs, to make use of their abilities (*Social Goals Generator*). *Deliberative* and *Social Goals Generators* save their goals in a *Goal* set and the agent cyclically checks the validity of every active goal.

To achieve the generated *goals*, agents outline *plans*. They consist of an ordered set of actions to be executed and some particular parameters to help the *Scheduler* organize and combine the proposed *plans*. Once the Deliberative and/or the Social Planner have drafted a plan, it is added to the IVA's *Plan* set. From this moment on, a plan waits to be incorporated into the *scheduler agenda*, unless the IVA decides to eliminate it. *Planners* are also responsible for the maintenance of some coherence between deliberative/social processes and the reactive ones, by updating the *concerns*.

Finally, the COGNITIVA *Scheduler* takes the actions proposed by the reactive, the deliberative and the social layers, scheduling them to be executed by agent' effectors.

### 3 Social Layer

The social layer has been designed to deal with the IVA's social capacity. It uses and updates the same information structures that are accessed by the other two layers. It receives the interpretation of the perceptions provided by the *Interpreter* and provides the *Scheduler* with the selected actions that must be executed. Here, we will describe the new features that provide COGNITIVA with effective social capabilities.

#### 3.1 Social Personal Model

Concerning the social layer, one of the agent's DC is the *Roles Set*, containing the services and abilities it possesses, can execute and provide to other agents. This set contains the roles of the agent, i.e., the capacity, function or task it can perform within the system it belongs to. Thus, we have added this new subset to the *Beliefs* set previously described. In addition, agents may also know (not always necessary) the set of the roles present in the system. The knowledge about distributed roles, also part of DC, can be included by the IVA's designer or be acquired throughout the agent's interactions with others. This information enables the IVA to make plans involving other agents' abilities. It is managed as a new DC structure called *Distributed Roles Set*.

### 3.2 Goals and Plans at the Social Layer

In the deliberative layer, the knowledge needed by the IVA to generate plans to reach its goals is ‘inside’ it, i.e., it should have all the necessary resources to do its plans. However, in the context of the social layer it is possible that a goal is planned counting on the roles provided by other system’s agents, including the case in which it can only be reached through the participation of others. We have extended COGNITIVA with two ways of obtaining a plan involving other agents’ roles. The first one is when the agent already knows the roles distributed in the system and, thus, can include them in its plans. The agent *Distributed Roles Set* should contain roles that allow it to create at least one plan to reach the goal. We have called this *direct planning*. The second possibility is when the agent asks another agent to generate a plan for it, in this case, an *indirect planning*. This last approach makes possible agent learning and knowledge aggregation of other agents’ roles for future interactions, allowing the IVA itself to generate similar plans in the future.

Both at COGNITIVA’s deliberative and social layers, it is foreseen that some unexpected situations may cause the re-planning of some of agent’s goals. In the social layer, this re-planning is more critical, given that the IVA may be too dependent on others to achieve its goals. When an agent re-plan one goal it generates a new sequence of actions until the plan succeeds or it decides that the plan is not appropriate/possible anymore. The main reasons why an action that is part of one of an agent’s plan needs to be *re-planned* are: (i) the provider agent of the failed *action* is not active/available at the moment; (ii) the requested action has less priority than the actions the provider agent is executing; and (iii) *action’s* expiry time has been achieved.

If the problem is a provider operation failure and the IVA cannot look for another provider, it cannot do much but to wait the provider to be active again. However, if the problem is priority or expiry time, we have included some possible approaches: (i) to increase the action priority every time the agent needs to re-request the action from the same provider; (ii) to force the priority of the action to be higher if there are few agents that can provide the role associated with this action; (iii) to increase the action expiry time, if the agent perceives that it will need to wait for the time the –maybe unique– provider establishes. The strategy to be employed is context-dependent. In addition, it should be expected that the IVA may not reach some of its goals.

At the time of generating plans for its goals the agent may not know which agents are able to provide it with the roles needed. However, the agent is able to look for an IVA that can be a provider of such roles. We have included in the *Social Reasoner* (an existing structure in the previous version of COGNITIVA) the charge of locating the role provider. Although approach choice is context-dependent, the most common ones for this task are: (i) Blackboard, where the agent that needs a role publishes its requests, (ii) Yellow Pages, where an agent announces the roles it can provide. Finding a role provider, however, does not guarantee its acceptance of collaboration.

As we have said, there is a strong influence of the IVA’s emotional state on its behavior. Concerning the social layer, this influence can be even more noticeable since agents may have more opportunity to choose among multiple actions and interactions, i.e., at the time the agent selects an interactive strategy, the *personal model* exerts a decisive pressure. As an example, if an IVA needs the execution of a specific role, it may be conservative and look for the same IVA that has previously provided it with

the role, or can be more intrepid and decide it wants to find out new role providers. If the previous interaction with a provider was not satisfactory, one agent can decide to change its plans in order to avoid a new interaction with that provider. The evaluation of an agent provider will be described later.

Actions generated by the social layer have always less priority than the reactive ones. However, there is no *a priori* priority difference between the deliberative and the social layer actions. Once again, *personality traits* and *moods* will influence on priority definition, to decide if the IVA executes first a social or a deliberative action.

### 3.3 Evaluating Social Interaction

The social layer allows an IVA to evaluate the other IVAs it interacts with. The more interaction a system promotes, the more an agent can evaluate others and, thus, delineate an *attitude* towards them. Although it seems to be a rich system mechanism, it must be optional, since we may face systems where evaluation is not important or may be useless and time consuming. At the current stage of our development, we are only considering *direct evaluation* between agents, i.e., evaluation from direct interaction between two agents.

One evaluation an IVA makes of others may be related to role confidence. This internal evaluation, although may be interesting in some cases, may be also difficult to achieve, given that it presupposes that the IVA has a way of comparing the role provided with a pre-defined 'product quality value'.

Another kind of evaluation that can be done is related to service quality: how a provider agent answers to the agent's requests. The ultimate intention is to use this information to decide which provider the IVA should choose the next time it needs the same role execution. We have defined some parameters that can be used to evaluate service quality: how many times a request has been sent before the role has been provided, how long a provider has been offering a role to an agent, how much time is spent between a request and the provider's answer, does it overpass the mean time to service provision?, just to mention some of them.

Whatever evaluation is done, it will influence on the agent behavior and on its interaction strategies within the system.

### 3.4 Social Contexts

The inclusion of the social layer makes possible to apply the architecture to a great amount of applications and, more importantly, to diversified contexts. The architecture can be used in applications requiring cooperation, information exchange, coordination and, also, in environments where cooperation gives rise to negotiation or competition. This diversity affects the contextualization of the architecture regarding the possible interactions among IVAs: (i) agents may always cooperate, (ii) agents may say no to cooperation, (iii) agents may want to negotiate, (iv) there may be uncertainty about the behavior of the agent.

Each of these aspects entails an increasing complexity, since the less the possibilities of executing successfully the agent's plans, the more intricate will be its behavior:

- If a provider agent always cooperates, it is more likely that a plan is concluded successfully, without re-planning demands;

- If there is a possibility that a provider agent does not cooperate, either because it is busy, unavailable or for any other reason, the probability that a plan fails is high. This situation will demand that the agent has enough flexibility or forecast ability to elaborate alternative plans in case of failure;
- If the environment demands negotiation, the required flexibility could be even greater, since the agent will have to evaluate the benefits and losses that it will have to manage during interaction;
- If there is complete uncertainty, the agent will have to be able to handle all the previous situations.

Up to now, we have concentrated on cooperative aspects and also on the possibility that agents do not always cooperate and the IVA has to re-plan its goals. Once we have obtained proper results, we will extend the model to include negotiation and competition. As a summary, Figure 1 illustrates the internal representation of COGNITIVA, in which its layers and their intra-relations can be seen.

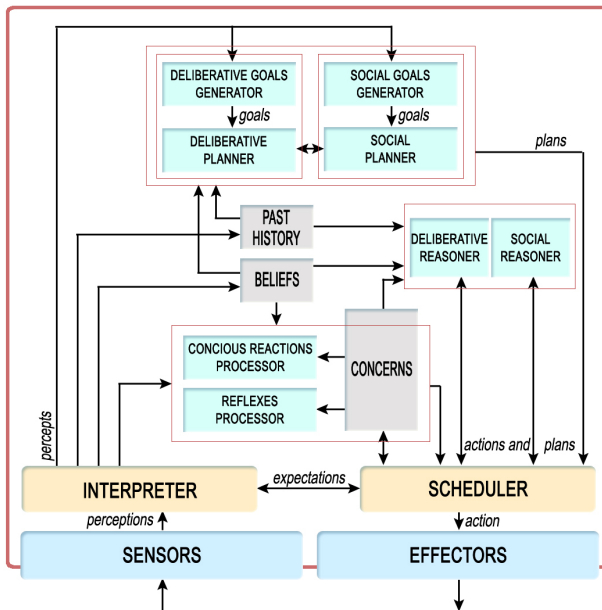


Fig. 1. Internal representation of COGNITIVA

## 4 Social Simulation

The application context in which we are testing the extension of COGNITIVA, with the described social layer, is a predation simulation environment. Predation is considered one of the most important selective pressures on wild animals, which must detect threats before they cause damage to themselves or to their offspring [23], [24], [25]. Before we offer system description and experiment parameters, we will present some theoretical background that motivates us to use this simulation environment.

To avoid being predated, an animal (prey) must be capable of early predation detection. Usually, an animal may detect threats on its own (individual vigilance) or may depend on signals from its associates (collective detection). According to [25], individual vigilance is more reliable and more rapid in threat detection but conflicts with other important activities, such as feeding, resting, grooming, fighting, mating, etc [23], [24]. Finding a way to manage this conflict is one of the reasons why many animal species aggregate. [25], [26] and [27] argue that the probability of being killed by a predator decreases with increasing group size, since associates can reveal they have detected a predator through escape behavior or by emitting recognizable alarms.

According to [25], great part of aggregation studies is related to anti-predator vigilance that, in turns, emphasize predator avoidance over other scanning functions, such as food search, moving planning (e.g., to survey escape routes) and social learning, because predation risk is said to be one of the most important factors that shapes animal vigilance. Many researches concerning vigilance centers on the conflict between feeding and vigilance, because many animals need to lower their heads to eat and thereby it drastically narrows their visual field. Even those that can feed in an upright position or eat using their hands may still need to concentrate their attention on finding, harvesting, or processing their food [25].

All these arguments emphasize the importance that vigilance has among animal activities. For our simulation purposes, we are going to concentrate on collective vigilance. Collective vigilance lies in the alternation of the individual animals' vigilance role to benefit the group, especially to solve vigilance conflicts with other essential activities. Based on many previous works, [25] affirms that "in cooperative groups, individuals may take specialized roles as sentinels [...] their associates then monitor the sentinels, rather than surveying the surroundings themselves". This kind of cooperation is usually called "reciprocal altruism", in which one organism provides a benefit to another in the expectation of future reciprocation and which utmost objective is specie preservation.

Within a group, collective vigilance can diminish individual monitoring, but demands that animals evaluate and estimate the number and kind of activities of their associates before abandoning monitoring to dedicate themselves to other activity. This kind of interaction between agents is part of an associate's social learning. Ethological references indicate that there are a lot of factors that could influence vigilance behavior. The most known are: age (closely related to previous experience), sex, reproductive status, social rank or special female status, such that of mothers with infants. Other factor is related to associates' monitoring (within-group vigilance) to avoid food theft or conspecific threat, although they are still quite unknown. At this stage of our work, we have not included such factors yet, mostly because they could add unknown complexity and cause incompressible variations. Food theft, for example, is clearly a competition aspect that we do not want to merge, at this point of our research, with cooperation analysis. We expect that they can be included in subsequent experiments, when some conclusions have been already made and some parameters have been controlled.

Once we eliminated all the variables that could make it difficult to analyze the outcomes, we are currently working on our simulation. The experiment consists in the interaction between a group of virtual animals, in this case, zebras, which can eat,

rest, wander and, more importantly from the social perspective, act as sentinel. Zebras' main objective is early predator detection through environment scan. In addition, zebras' individual goal is to maintain their energy reserves at a desired level, as well as their *emotional* parameters. As we could learn from theory, the most important emotion in the context of our experiment is *fear*. This is crucial to drive zebras to look for a secure state.

Thus, the purpose of the experiment is *to find out if collective vigilance maintenance can emerge from the (indirect) variation of emotional parameters (fear) and cooperative traits in virtual animals*, i.e., if it is possible to simulate that, at least one of them will assume a sentinel role at a given moment, without an explicit definition of a minimal value for quantity of sentinels within the system. To achieve it, we have considered [28]'s arguments, that an animal will decide if it is going to assume a sentinel role based on two variables: (i) Its energy reserves: although vigilance is crucial for survival, a high level of hunger may switch animal's priorities. So, in case that the hunger level is above an upper bound value, an animal will not act as a sentinel, and it will prefer to look for, find and process food; (ii) Its associates' actions, i.e., what the others are doing. Knowing if any of them is already monitoring predators is decisive to determine an animal's own action.

These two variables are intertwined: an animal assumes a sentinel role if its energy reserves are greater than a lower threshold that, in turn, varies depending on its associates' current role. In our simulation, this relation is defined as a variation of the physical parameters (*hunger*) threshold related to the perceived insecurity; each animal might have different threshold values. We also have defined that no-vigilance increases insecurity and thus IVA's *fear*: it will influence on the animal decision to assume or not a sentinel role. In addition, we will vary the animal's time interval to scan associates (to find if there are some of them watching over). To obtain this result, we have made some necessary assumptions to block undesired behavior. We assume that: (i) There will always be enough food for animals and they will not fight for it. The purpose is to prevent conspecific competition or within-group vigilance; (ii) Animals can be constantly eating or resting (its preferred activities). Our objective is to prevent that changes in an animal role (to become a sentinel) be based on IVA's free time. Thus, an animal monitors predator threats if it decides to, not because there is nothing else better to do.

One possible evaluation approach in our virtual scenario is to allow an IVA to evaluate the amount of time each virtual animal has spent as sentinels, compared to other agents' time. It may mean, for example, that an animal is not as cooperative as others or may mean that the animal is not so good in playing the sentinel role or is thoughtless about its importance. In this first stage of the evaluation of our proposal, we have not included evaluation among animals. Although we have included differences in virtual animals' *personality traits* that may cause more or less cooperation, animals are not aware of them and they assume that all cooperate equally.

Some data we are collecting and analyzing are: (i) the number of times each animal has played the role of sentinel; (ii) the time each one has dedicated to this role; (iii) what are the reasons leading animals to decide to abandon the sentinel role; (iv) the number of sentinels along the experiment; (v) how aggregate were agents; (vi) the number of simultaneous sentinels along interactions. Besides evaluating our objective we expect that the simulation will allow us to evaluate some other results, such as if

IVAs configured as more cooperative have dedicated more time acting as sentinel than the others, or if the agents have a tendency to be aggregated.

## 5 Final Comments and Ongoing Work

Nowadays, there are few emotion-based architectures for controlling IVAs behavior that consider their social dimension. The results COGNITIVA has attained for managing individual behavior and its availability to consider social abilities are a good incentive to extend it with a social layer, contributing to fill that gap.

The new social dimension of COGNITIVA will be tested through the evaluation that we are currently developing and that has been described above. We expect that the evaluation will allow us to come up with conclusions related to the influence that COGNITIVA's emotional parameters have on individual behavior, as well as on the general system behavior.

As future work, we intend to include a number of parameters we have mentioned above, but have not included in the system yet, mainly those related to competition, negotiation between agents. We also plan to use COGNITIVA to simulate other social environment that were previously simulated in a non-emotional agent-oriented architecture. We believe that it will be useful not only to compare architecture performance but also as an extra validation proof for our research.

**Acknowledgments.** João Queiroz is sponsored by FAPESB/CNPq. Jackeline Spinola and João Queiroz would like to thank the Brazilian National Research Council (CNPq) and The State of Bahia Research Foundation (FAPESB).

## References

1. Damásio, A.R.: Emotion and the Human Brain. *Annals of the New York Academy of Sciences* 935, 101–106 (2001)
2. Freitas, J.S., Gudwin, R.R., Queiroz, J.: Emotion in Artificial Intelligence and Artificial Life Research: Facing Problems. In: Panayiotopoulos, T., Gratch, J., Aylett, R., Ballin, D., Olivier, P., Rist, T. (eds.) *IVA 2005. LNCS (LNAI)*, vol. 3661, Springer, Heidelberg (2005)
3. McCauley, T.L., Franklin, S.: An architecture for emotion. In: *Proceedings of the 1998 AAAI Fall Symposium. Emotional and Intelligent: The Tangled Knot of Cognition*, pp. 122–127. AAAI Press, CA (1998)
4. Ray, P., Toleman, M., Lukose, D.: Could emotions be the key to real artificial intelligence? *ISA'2000 Intelligent Systems and Applications*, University of Wollongoo, Australia (2001)
5. Cañamero, D.: Modeling Motivations and Emotions as a Basis for Intelligent Behavior. In: *First Conference on Autonomous Agents*, pp. 148–155. ACM, California (1997)
6. Damásio, A.R., Grabowski, T., Bechara, A., Damásio, H., Ponto, L.L., Parvizi, J., Hichwa, R.D.: Subcortical and cortical brain activity during the feeling of self-generated emotions. *Nature Neuroscience* 3(10), 1049–1056 (2000)
7. Ledoux, J.: *The emotional brain: the mysterious underpinnings of emotional life*. Touchstone, New York (1996)
8. Nesse, R.M.: Computer emotions and mental software. *Social neuroscience bulletin* 7(2), 36–37 (1994)

9. Imbert, R.: Una Arquitectura Cognitiva Multinivel para Agentes con Comportamiento Influido por Características Individuales y Emociones, Propias y de Otros Agentes. Ph.D. Thesis, Computer Science School, Universidad Politécnica de Madrid (2005)
10. Imbert, R., de Antonio, A.: When Emotion Does Not Mean Loss of Control. In: Panayiotopoulos, T., Gratch, J., Aylett, R., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 152–165. Springer, Heidelberg (2005)
11. Imbert, R., de Antonio, A.: An Emotional Architecture for Virtual Characters. In: Subsol, G. (ed.) ICVS 2005. LNCS, vol. 3805, pp. 63–72. Springer, Heidelberg (2005)
12. Keltner, D., Kring, A.M.: A. M. Emotion, Social Function, and Psychopathology. *Review of General Psychology* 2(3), 320–342 (1998)
13. Averill, J.R.: A constructivist view of emotion. In: Plutchik, R., Kellerman, H. (eds.), pp. 305–339. Academic Press, New York (1980)
14. Campos, J., Campos, R.G., Barrett, K.: Emergent themes in the study of emotional development and emotion regulation. *Developmental Psychology* 25, 394–402 (1989)
15. Ekman, P.: An argument for basic emotions. *Cognition and Emotion* 6, 169–200 (1992)
16. Lazarus, R.S.: *Emotion and adaptation*. Oxford University Press, New York (1991)
17. Frijda, N.H., Mesquita, B.: The social roles and functions of emotions. In: Kitayama, S., Marcus, H. (eds.) *Emotion and culture: Empirical studies of mutual influence*, pp. 51–87. American Psychological Association, Washington (1994)
18. Ohman, A.: Face the beast and fear the face: Animal and social fears as prototypes for evolutionary analysis of emotion. *Psychophysiology* 23, 123–145 (1986)
19. Clark, C.: Emotions and the micropolitics in everyday life: Some patterns and paradoxes of Place. In: Kemper, T.D. (ed.), pp. 305–334. New York Press, Albany (1990)
20. Galvão, A.M., Barros, F.A., Neves, A.M.M., Ramalho, G.: Persona-AIML: An Architecture for Developing Chatterbots with Personality. In: Jennings, N.R., Sierra, C., Sonenberg, L., Tambe, M. (eds.) *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, pp. 1264–1265. ACM Press, New York (2004)
21. Charles, F., Cavazza, M.: Exploring Scalability of Character-Based Storytelling. In: Jennings, N.R., Sierra, C., Sonenberg, L., Tambe, M. (eds.) *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, pp. 870–877. ACM Press, New York (2004)
22. Delgado-Mata, C., Aylett, R.: Emotion and Action Selection: Regulating the Collective Behaviour of Agents in Virtual Environments. In: Jennings, N.R., Sierra, C., Sonenberg, L., Tambe, M. (eds.) *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, pp. 1302–1303. ACM Press, New York (2004)
23. Dimond, S., Lazarus, J.: The problem of vigilance in animal life. *Brain, Behavior and Evolution* 9, 60–79 (1974)
24. Mooring, M.S., Hart, B.L.: Costs of allogrooming in impala: distraction from vigilance. *Animal Behaviour* 49, 1414–1416 (1995)
25. Treves, A.: Theory and method in studies of vigilance and aggregation. *Animal Behaviour* 60(6), 711–722 (2000)
26. Bednekoff, P.A., Lima, S.L.: Randomness, chaos and confusion in the study of antipredator vigilance. *Trends in Ecology and Evolution* 13, 284–287 (1998)
27. Dehn, M.M.: Vigilance for predators: detection and dilution effects. *Behavioral Ecology and Sociobiology* 26, 337–342 (1990)
28. Bednekoff, P.A.: Coordination of safe, selfish sentinels based on mutual benefits. *Annales Zoologici Fennici* 38, 5–14 (2001)



# Just-in-Time Monitoring of Project Activities Through Temporal Reasoning

Sara E. Garza and José Luis Aguirre

Tecnológico de Monterrey  
Centro de Sistemas Inteligentes  
{A00592719,jlaguirre}@itesm.mx

**Abstract.** A critical issue in project management is time and its administration. With the introduction of autonomous or semi-autonomous systems that handle tasks related to projects, time management can become even more complex. In this paper, a temporal reasoning based mechanism for monitoring the execution times of activities that are developed during a project is being proposed, with the intent of detecting temporal discrepancies (delays or activities going ahead of time). The mechanism relies on the use of Allen's interval algebra and allows uncovering concurrent relationships, which are useful to consider for making the necessary adjustments to the project when discrepancies are detected. This can be especially convenient in automated or semi-automated contexts, where activities are assigned to intelligent agents. With this in mind, the mechanism was integrated to the JITIK multiagent system. It has been tested with representative cases and has been found to detect all the situations where delays are involved.

## 1 Introduction

Time is a critical constraint to be considered in the management of a project, since it can determine success or failure depending on how well it is administered. Therefore, processes such as time monitoring become important. Moreover, there are contexts (i.e manufacturing and teamwork supported by electronic personal assistants) in which some or all of the project management tasks are automated with the aid of intelligent systems. Having this in mind, in this paper a temporal reasoning based mechanism for monitoring the execution times of activities that are developed during a project, which is suited to be integrated with a multiagent based knowledge and information distribution system, is being proposed. In this case, the integration to the JITIK system (that stands for Just-In-Time Information and Knowledge), which is a multiagent solution whose primary intention is bringing just-in-time information to interested users inside an organization, and works by examining various data sources [8] is considered. The proposed mechanism tries to detect temporal discrepancies in the form of delays or activities going ahead of time, and notify their existence.

The present work is organized as follows: in Section 2, a general background that covers the main themes related to the research is provided; in Section 3,

the monitoring mechanism is introduced; in Section 4, some of the designed test cases for observing the mechanism's behavior are explained; and in Section 5 the most relevant conclusions of the work are drawn.

## 2 Background

The designed monitoring mechanism relies on three basic areas: Temporal Reasoning, project management, and multiagent systems (specially JITIK). Therefore, a broad view of each one of these topics will be given.

Temporal Reasoning (TR) consists of formalizing the notion of time and providing means for representing and reasoning about temporal aspects of knowledge [14]. Allen's Interval Algebra [1] and Freksa's semi-intervals [6] are of interest.

Interval algebra works over 13 basic temporal relationships (Table 1) and defines addition and multiplication operations. Temporal inferences are done via multiplication in an interval net, which is a graph where nodes are intervals and arcs are sets of relations (relation vectors); addition is used when updating the net [15]. With respect to semi-intervals [6], these consist of intervals that have only one defined endpoint and are used for working with incomplete or uncertain information.

Regarding project management, the concepts that were used deal with activity networks and the Critical Path Method (CPM). An activity network depicts the workflow of a project. It consists of an acyclic digraph where nodes represent activities and arcs portray dependency relationships among them [4]; this representation is known as Activity-On-Arrow or AOA [12]. Finish-to-Start dependencies mean that the end of one activity is tied with the beginning of another one. Furthermore, the CPM [12] is a workflow analysis technique that allows detecting activities that can be postponed certain time without affecting the project's development, and detection of activities that should be executed within specific time points. The Critical Path (CP) is an activity chain that by definition has no slack. Some important concepts involving the CPM method are the milestones (revision points), early and late start dates, early and late finish dates, and slack or float [9].

The purpose of the JITIK multiagent system is to deliver the correct *information* to the correct *persons* at the correct *time* in an organization [8]. Regarding its multiagent model [3], it consists of various types of agents. First, there are Personal Agents that work on behalf of the members of the organization by filtering and delivering useful content according to user preferences. The Site Agent provides IK to the Personal Agents, acting as a broker between them and Service agents. Service agents collect and detect IK pieces that are supposed to be relevant for someone in the organization [11]. That knowledge is hierarchically described in the form of taxonomies. The model is described in Figure 1.

**Table 1.** Temporal Relationships in Allen's Interval Algebra

Relation	Symbol	Inverse	Illustration
<i>X before Y</i>	<i>b</i>	<i>a</i>	XXX YYY
<i>X equal Y</i>	<i>e</i>	<i>e</i>	XXX YYY
<i>X meets Y</i>	<i>m</i>	<i>m-</i>	XXXYYY
<i>X overlaps Y</i>	<i>o</i>	<i>o-</i>	XXX YYYY
<i>X during Y</i>	<i>d</i>	<i>d-</i>	XXX YYYYY
<i>X starts Y</i>	<i>s</i>	<i>s-</i>	XXX YYYYY
<i>X finishes Y</i>	<i>f</i>	<i>f-</i>	XXX YYYYY

### 3 Monitoring Mechanism

The mechanism works as follows: the monitoring task is performed in certain time points and the project is inspected in order to find temporal situations that are out of order; if that happens, notifications are sent according to the JITIK scheme. The monitoring task contrasts a theoretical schema (obtained via TR) against the actual one that is taking place. According to the project's progress, actual relations happening are calculated; each relation (actual schema) is compared with a correspondent inferred relation vector (ideal schema). If the relationship is not a member of the vector, a temporal discrepancy is notified. If there were no discrepancies, the inferred relation vector is updated to reflect time changes and maintain consistency. The JITIK scheme was adjusted by introducing activity agents, which are assigned one or more project activities. If a delay is detected, the agents can communicate, make notifications, and negotiate with other agents affected by temporal changes to the project.

#### 3.1 Elements of the Mechanism

The main elements of the mechanism are the same used within the domain of project management, but they have been abstracted to fulfill the mechanism's function.

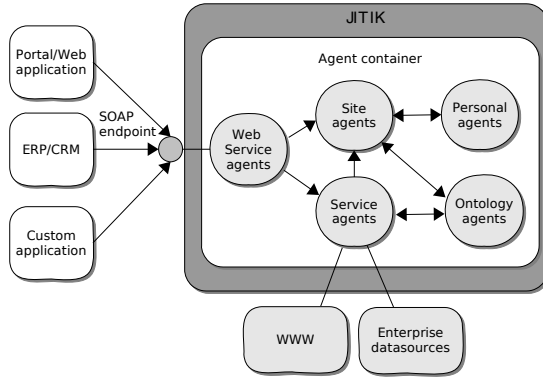


Fig. 1. JITIK agents

**Project.** Inside the created framework, every project is viewed as a structure that has five key components: 1) a starting point and a finishing point in time, 2) a duration, 3) a set of monitoring points, 4) a set of milestones, and 5) a set of interrelated activities.

Formally, a project can be defined as a tuple

$$P = (F, PT, dp, G, M, PM)$$

where:

**F** is the *set* of dates involved in the project execution; **PT** is the *set* of time points involved in project execution (obtained from F); **dp** is the project *duration*; **G** is a *graph* that represents the activity network; **M** is the *set* of milestones,  $M \subset PT$ , and **PM** is the set of monitoring points,  $PM \subseteq PT$ .

With respect to G, it has the properties of a PERT digraph [5] defined as  $G = (A, E)$ . A represents the set of activities to be executed during the project, and E represents the precedence relationships among them. Now, due to the fact that an activity network portrays activities and dependencies that hold among them, it becomes possible to overload G in order to depict an interval net at the same time. In that way, a project’s workflow can be visualized as a field where TR may be used.

**Activities.** Activities themselves can be considered as time intervals, since they possess a duration and—consequently—have endpoints. In fact, they can be seen as sub-intervals that take place inside a project, which is a larger interval. Formally, every activity can be defined as a tuple

$$x = (id, name, state, d, es, ef, ls, lf, fl, cp, rs, rf, pr, su)$$

where:

**id** is an activity identifier; **name** is a descriptive label for the activity; **state** is the state of progress (Not Started, Started, Finished); **d** is duration; **es** is the

early start time; **ef** is the early finish time; **ls** is the late start time; **lf** is the late finish time; **fl** is the float (slack); **cp** determines if the activity belongs to the critical path; **rs** is the real start time; **rf** is real finish time; **pr** is the set of predecessors; **su** is the set of successors.

Note that  $x \in A$

**Relationships.** Two kinds of relationships between pairs of activities are defined: direct and inferred. Direct relationships are shown by the workflow’s arcs; inferred relations portray concurrence relationships (concurrent means that the vectors contain other relations besides  $a, b, m$ , and  $m-$ ) and are derived by applying TR to the existing direct links, and they can be divided into relevant and irrelevant. Irrelevant relations do not provide substantial information for determining temporal discrepancies, either for being too “wide” (non restrictive) or for being a useless precedence relation (remote precedence); i.e., a vector with the 13 relations does not constrain how an activity will be developed with respect to another, and knowing that the first activity goes before the last one is also futile. Now, for the monitoring process, only relevant relationships are of interest, because they portray the temporal options that should take place between activities in order to keep the project on time. In that sense, they provide restrictions just as direct relationships do; the difference consists in the fact that relevant relations can link intervals that may not seem to have a connection.

With the purpose of formalizing this notion, the following sets are introduced.

$$\begin{aligned}
 Allen &= \{a, b, d, d-, e, f, f-, m, m-, o, o-, s, s-\} & Allen_{CP} &= \{a, b, m, m-\} \\
 Allen_{NCP} &= \{m, m-\}
 \end{aligned}$$

Therefore, the types of relationships defined above can be formalized as it is now suggested:

**Direct relationships**

$r = (a1, d, a2)$ , where:

- $d \in Allen_{CP}$  if it only relates activities pertaining to the CP
- $d \in Allen_{NCP}$  if at least one activity does not belong to the CP

**Inferred Relationships**

$r = (a1, i, a2)$ , where:

- $i \subset Allen$  if it is relevant
- $i = Allen$  if it is non restrictive irrelevant
- $i \subset Allen_{NCP}$  if it is irrelevant of remote precedence

**3.2 Monitoring Procedure**

The main procedure used to monitor the activities, RT-MONITOR is now discussed; the pseudocode of this algorithm is located in Figure 2.

```

function RT-MONITOR (activity, t) returns void
inputs: activity, activity for which monitoring is executed
         t, time point in which monitoring is done

Relevant  $\leftarrow$  set of relevant relations activity

if Relevant =  $\emptyset$  //There are no relevant relations activity
  Compare actual state of activity vs. state that it should have given t
  if there exist discrepancies between states
    DO-NOTIFICATION({activity})
  else if there are no discrepancies, pero activity has not started
    Predecessors  $\leftarrow$  predecessor activities of activity
    for each pred  $\in$  Predecessors
      RT-MONITOR (pred, t) // Check recursively each predecessor

else // There exist relevant relations
  for each relation  $\in$  Relevant
    // According to the state of both activities in the relation,
    // take course of action
    if activities have not started
      check if activities should have started already
      if activities should have started
        DO-NOTIFICATION({activity, relact})
      else
        Predecessors  $\leftarrow$  activity predecessors
        for each pred  $\in$  Predecessors
          RT-MONITOR(pred, t)
    else if any of the activities has started
      actual  $\leftarrow$  current relation between activities
      if actual  $\in$  Relation
        relation  $\leftarrow$  updated relation vector
      else
        DO-NOTIFICATION({activity, relact})
    else
      if any of the activities is ahead
        DO-NOTIFICATION ({activity, relact})

```

**Fig. 2.** Monitoring mechanism main procedure

**Previous steps.** Before running the monitoring procedure, activity and project data are captured, and inferred relation vectors are obtained; this last step is done with the TimeGraph II system [7]. Once this has been completed, at each established monitoring point the mechanism is executed, and the monitoring is made with respect to the CP activity that is nearest to the next revision point (“milestone activity”).

If the activity has no relevant (concurrent) relations, then its current state is just contrasted with the state it should have to the present date. If the states differ, a notification is sent. On the contrary, if the activity does contain relevant relations, one of several courses of action—that depend on the progress state of the two related activities—is taken.

When none of the activities has started, the course of action consists of checking the *ls* times to see if this situation should be happening. If they should have already begun, a notification is sent; otherwise, the monitoring procedure is called recursively. This allows the mechanism to inspect tasks that are to be executed before the milestone activity. On the other hand, if at least one of the the activities has already started, the course of action basically consists of calculating the actual relationship between the intervals with their available endpoints. Once this relation is estimated, the inferred relation vector is loaded, and the relation is searched within the inferred set. If the relationship is not found, this means that one (or maybe both) of the activities presents a temporal discrepancy; consequently, the next step that the mechanism takes is to determine which activity is the one that shows variations with respect to the planned scheme and what type of discrepancy is happening (the activity is delayed, the activity goes ahead of time). When this is known, a notification is sent to the pertinent entities involved with the activity. Now, if both activities have ended, the course of action consists of checking whether one or the two activities went ahead of time. If this is the case, a notification is sent.

In order to determine those entities that might be interested in knowing that a temporal discrepancy has taken place, several considerations have to be taken into account. For logical reasons, the first ones that are impacted by discrepancies are the *direct successors* of the activity. Furthermore, agents belonging to *concurrent activities* are also affected by the delay or work ahead of another task, since this can modify their slack. Finally, all of the *agents assigned* to the activity must be informed, regardless of whether the activity is their main task or a subtask (for example, an agent related to a design task is concerned with every activity that belongs to design). In consequence, the mechanism gathers these agents (that is, it determines who they are depending on the activity) for sending them the notification.

Concerning the notification scheme, as it was stated previously, the JITIK format is used for notifying, since the mechanism was built to cope with this system. However, it is necessary to make some adjustments to the original format for adapting it to the context. This is done by introducing *activity agents*, which have project activities assigned and are similar to JITIK personal agents. Activity agents can be assigned one or more activities.

Finally, in order to maintain consistency (since some relations will become unfeasible as time passes), the inferred vectors must be updated. This is done by calculating semi-intervals over the existing activity data; the set of obtained semi-intervals is contrasted against the relation vectors, and only those relationships that are contained in both sets are kept.

## 4 Tests

With the purpose of observing the mechanism’s behavior and showing if it achieved its main objective—detecting temporal discrepancies in a project’s activities—, a sample workflow is presented as a test case. Figure 3 portrays the activity network that belongs to the presented scenario, and Table 2 shows the most important data of each activity.

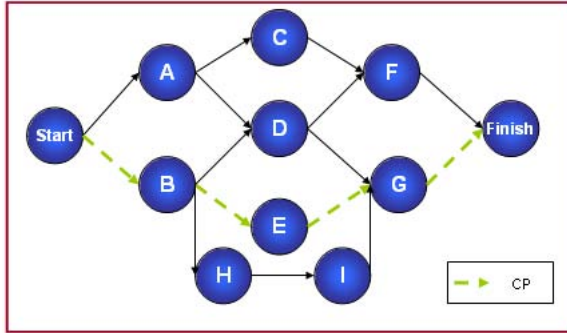


Fig. 3. Test case activity network

Because there are many possible situations where discrepancies can arise, only four basic test cases were selected out of this scenario: 1) a critical path activity suffers a delay, 2) a non-critical activity suffers a delay, 3) a critical path activity goes ahead of time, 4) a non-critical activity goes ahead of time. These tests were drawn from two main criteria: belonging to the critical path (yes/no) and type of discrepancy (delay, going ahead). In addition, Case 3 presents a situation where several discrepancies happen; it was chosen with the purpose of showing if the mechanism can cope with this kind of events.

Table 2. Scenario data. Act.–activity label, d–duration, es–early start time, ls–late start time.

Act.	d	es	ls
A	3	0	10
B	11	0	0
C	5	3	19
D	10	11	13
E	12	11	11
F	8	21	24
G	9	23	23
H	5	11	14
I	4	16	19



Also, it is important to note that a milestone was set in time point 23 (which is reasonable, since this point is near the middle of the project) and is relevant to recall that every activity is assigned to at least one agent.

Initially, in the database all the activities' states are set to NS (Not Started), and thus their real start and finish times have no significant values. However, for each test case, different parameters were set up in order to carry it out. These are depicted in Table 3.

**Table 3.** Test case parameters. S=started, F=finished.

Case	Description	Parameters
1	B delayed	A: state=F; B: state=S; monitoring point=8
2	E ahead, H and I delayed	A, B, E: state=F; C, D: state=S; mon. point=19
3	H delayed	A, B: state=F; C, D, E: state=S; mon. point=8
4	A ahead	A: state=F; B: state=S; mon. point=10

**Table 4.** Test results

Case	Discrepancy detected	Activities to notify
1	Yes	A, C, D, E, H
2	Yes	A, D, F, E, G, H
3	Yes	E, I
4	No	None

**Results.** Table 4 summarizes the results obtained by running the mechanism with the test scenario previously mentioned. The *Case* column provides identifiers for the test cases of the scenario, the *Discrepancy* column tells if the temporal discrepancy was correctly detected (test passed or failed), and the *Activities to notify* column depicts the activities that will be notified about the discrepancy.

According to the presented scenario, the mechanism was able to deliver the expected results in all cases where delays were present. However, there are instances of non-critical path activities going ahead of time that it cannot currently detect. These arise when a discrepancy concerning solely a change in duration, results in a relation that is contained actually in the inferred vector; then, the discrepancy can not be detected. Nevertheless, it can also be seen that the mechanism also detects simultaneous discrepancies at a time. Additionally, it has been able to yield consistent results within the applied test cases, meaning that it works correctly (i.e., activities with an “on-time” status have not been mistakenly regarded as having discrepancies, agents are accurately identified, etc.).

## 5 Conclusions and Related Work

A TR-based mechanism for monitoring the temporal execution of the activities involved in the development of a project, which is suited for knowledge and

information distribution multiagent-based systems (specifically, JITIK) has been presented. The mechanism uses Allen's interval algebra, and currently works with AON activity networks of Finish-to-Start mandatory dependencies. The mechanism has been tested and it has detected all delay cases that were present in the test set. The advantage of this approach is that it considers activities as interrelated entities, not isolated items, enabling to uncover concurrent relations that may not be visible at a first glance, especially in large projects. This can be convenient in certain environments where project management tasks are assigned to intelligent agents, because if a discrepancy is detected, the agents can work together trying to rearrange the development of the project.

It is relevant to point out that it was found that the works that verse over this context do not employ TR in the way that it is used in this research. In [2], a framework for monitoring elderly persons in intelligent houses is described; they gather relevant data (the person's activity level, location, if there are appliances turned on in the house, etc.) and determine whether the actions the person does are normal or if the person is in some kind of danger. Temporal reasoning is utilized in the form of ECA (Event-Condition-Action) rules that are fired given a situation that occurs on a specific event. The approach of TR that is being used is more oriented to temporal semantics, which is devoted to finding relations among points of time inside a specific context [13]. In that sense, this kind of "reasoning" is more qualitative—unlike the temporal algebras. In [10] the RASTA system, which performs temporal abstractions (these include various types of inferences) over large sets of medical data, is described. The difference with respect to the present research is the TR approach used. While the mechanism is based in Allen's interval algebra, this system is based on a framework created for deriving temporal properties given isolated pieces of information. In this case, besides working within the context of project management, Allen's interval algebra was utilized because the nature of the data—which is more quantitative—seems to cope better with this kind of approaches.

Future work includes, on the one hand, extending the mechanism to incorporate other types of activity networks and TR approaches. Among other benefits, this must improve the monitoring procedure to detect the cases that are actually missed. On the other hand, it would also be relevant to test it in real project management environments and incorporating it to the project management control process and to other systems. Finally, a complexity analysis could be done to validate the feasibility of its implementation.

## References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. *Communications of the ACM* (1983)
2. Augusto, J.C., Nugent, C.D., Black, N.D.: Management and analysis of time-related data in smart home environment. Technical report, University of Ulster at Jordanstown (2004)

3. Brena, R., Aguirre, J., Treviño, A.C.: Just-in-Time Information and Knowledge: Agent technology for KM Bussiness Process. In: Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference (2001)
4. Elmaghraby, S.E.: Activity Networks: Project Planning and Control by Network Models. John Wiley and Sons, New York (1977)
5. Even, S.: Graph Algorithms. Computer Science Press, Maryland (1979)
6. Freksa, C.: Temporal reasoning based on semi-intervals. Artificial Intelligence (1992)
7. Gerevini, A., Schubert, L., Shcaeffe, S.: The temporal reasoning systems timegraph i-ii. Technical report, Istituto per la Ricerca Scientifica e Tecnologica, University of Rochester, University of Alberta (1995)
8. ITESM CSI. Just In Time Information and Knowledge (2006), <http://lizt.mty.itesm.mx/jitik>
9. Morris, P.W.G., Pinto, J.K.: The Wiley Guide to Managing Projects. John Wiley and Sons, New Jersey (2004)
10. O'Connor, M., Grosso, W.E., Tu, S.W., Musen, M.A.: Rasta: A distributed temporal abstraction system to facilitate knowledge-driven monitoring of clinical databases. Technical report, Stanford School of Medicine (1995)
11. Pintero, R.F.B., Cervantes, J.L.A., Chesnevar, C., Luna, L.G.: Knowledge and information distribution leveraged by intelligent agents. Knowledge and Information Systems Journal (2006)
12. PMI. A Guide to the Project Management Body of Knowledge. Project Management Institute, Pennsylvania (2004)
13. Shahar, Y.: A framework for knowledge-based temporal abstraction. Technical report, Stanford University (June 1997)
14. Vila, L.: A survey on temporal reasoning in artificial intelligence. AI Communications 7 (March 1994)
15. Vilain, M., Kautz, H., van Beek, P.: Constraint propagation algorithms for temporal reasoning. In: Proceedings of the AAAI 1986 (1986)

# Scaling Kernels: A New Least Squares Support Vector Machine Kernel for Approximation

Mu Xiangyang<sup>1</sup>, Zhang Taiyi<sup>1</sup>, and Zhou Yatong<sup>2</sup>

<sup>1</sup> Dept. of Information and Commun. Engineering, Xi'an Jiaotong University,  
Xi'an 710049, China

muyou98@xjtu.edu.cn

<sup>2</sup> School of Information Engineering, Hebei University of Technology,  
Tianjin 300401, China

zhouyatong\_zw@126.com

**Abstract.** Support vector machines(SVM) have been introduced for pattern recognition and regression. But it was limited by the time consuming and the choice of kernel function in practical application. Motivated by the theory of multi-scale representations of signals and wavelet transforms, this paper presents a way for building a wavelet-based reproducing kernel Hilbert spaces (RKHS) which is a multiresolution scale subspace and its associate scaling kernel for least squares support vector machines (LS-SVM). The scaling kernel is constructed by using a scaling function with its different dilations and translations. Results on several approximation problems illustrate that the LS-SVM with scaling kernel can approximate arbitrary signal with multi-scale and owns better approximation performance.

**Keywords:** Least squares support vector machine, Scaling kernel, Reproducing kernel Hilbert spaces.

## 1 Introduction

The last several years have witnessed an increasing interest in support vector machine (SVM) [1][2]. Least squares support vector machine (LS-SVM) is a least squares version of SVM's. In this version one finds the solution by solving a set of linear equations instead of a time-consuming quadratic program (QP) for classical SVM's[3][4]. This is due to the use of equality instead of inequality constraints in the problem formulation. Although LS-SVM was developed initially for classification, it has been successfully extended to handle approximation.

The performance of LS-SVM largely depends on the kernel. There are many kinds of kernels such as polynomial kernel, radial basis function kernel (Gaussian kernel), and tangent distance kernel, *etc.*, can be used. Every kernel has its advantages and disadvantages. Unfortunately, there are currently no theories available to "learn" the form of the kernels. Among the possible kernels, the most used in practice are radial basis function kernel. Despite its widespread success, the radial basis function kernel suffers from the following two limitations. Firstly the radial basis functions can't form a set of complete bases, resulting in that the LS-SVM can't approximate arbitrary signal. Secondly, a multi-scale approximation provides a simple hierarchical

framework for interpreting the signals. However, the LS-SVM with radial basis function kernel can't implement multi-scale approximation.

Recently, there is a growing interest around wavelet kernel and multi-scale kernels. Zhang presents a wavelet support vector machine (WSVM) [5]. Opfer constructed a wavelet kernel associated with Sobolev spaces, a well-known RKHS [6]. Amato proposed the concept of multi-scale kernels [7]. Rakotomamonjy presented a method for constructing a RKHS and its associated kernel from a Hilbert space with frame elements having special properties[8]. Motivated by the theory of multi-scale representations of signals and wavelet transforms, this paper constructs a scaling kernel by using a scaling function with its different dilations and translations. The way we construct the scaling kernel is similar to the one Rakotomamonjy and Opfer described in the example. Compared to the radial basis function kernel, our kernel enjoys two advantages: (1) it can approximate arbitrary signal and owns better approximation performance; (2) it can implement multi-scale approximation.

The remainder of this paper is organized as follows. In the next section, we review the standard LS-SVM. Following that, in Sec.3 we illustrate how to construct the scaling kernels. Sec.4 reformulate the LS-SVM with the scaling kernel. Experiments are reported in Sec.5 and some conclusions are presented in Sec. 6.

## 2 Review of Least Squares Support Vector Approximation

In approximation problems, we are given a set of training data  $D = \{(x_n, t_n), n = 1, \dots, N\}$ , which is collected by randomly sampling an underlying signal  $f(x)$ . Approximation aims to recover the underlying signal  $f(x)$  from the finite data set  $D$ .

LS-SVM has been developed for solving above approximation problem. By some nonlinear mapping  $\varphi$ , the data set is mapped into the feature space  $F$  in which a linear approximation is defined

$$f(x) = \mathbf{w}^T \varphi(x) + b \tag{1}$$

where  $\mathbf{w}$  is weight vector and  $b$  is bias term. To obtain  $\mathbf{w}$  and  $b$  one solves the following constrained optimization problem

$$\min_{\mathbf{w}, b, e} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{2} \sum_{n=1}^N \xi_n^2 \tag{2}$$

subject to the equality constraints

$$t_n - \mathbf{w}^T \varphi(x_n) - b = \xi_n, \quad n = 1, 2, \dots, N \tag{3}$$

with  $\xi_n$  a error variable,  $C$  a regularization factor and  $\xi_n^2$  the least squares cost function.

One define the Lagrangian

$$\min G(\mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{2} \sum_{n=1}^N \xi_n^2 + \sum_{n=1}^N \alpha_n (t_n - \mathbf{w}^T \varphi(x_n) - b - \xi_n)$$

where  $\alpha_n$  are the Lagrange multipliers that can be either positive or negative due to the equality constraints. After taking the conditions for optimality, one obtain a linear system in the dual space

$$\begin{bmatrix} 0 & \mathbf{L}^T \\ \mathbf{L} & \boldsymbol{\chi}^T \boldsymbol{\chi} + C^{-1} \mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{T} \end{bmatrix}$$

where  $\mathbf{T} = (t_1, t_2, \dots, t_N)^T$ ,  $\boldsymbol{\chi} = (\varphi(x_1), \varphi(x_2), \dots, \varphi(x_N))$ ,  $\mathbf{a} = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$ ,  $\mathbf{L} = (1, 1, \dots, 1)^T$ , and

$$\mathbf{w} = \sum_{n=1}^N \alpha_n \varphi(x_n) . \tag{4}$$

Finally, we substitute Eq. (4) into Eq. (1) and arrive at

$$f(x) = \sum_{n=1}^N \alpha_n K(x_n, x) + b , \tag{5}$$

where the function  $K(x_n, x)$  is a positive definite kernel associated with the feature space  $F$ .

### 3 Building Scaling Kernels

Since the development of LS-SVM as a tool for signal approximation, there is a growing interest around RKHS. A RKHS is a Hilbert space of functions with special properties [9]. It plays an important role in approximation as it allows to write in a simple way the solution of a learning from empirical data problem.

Here we firstly sketch a way to build a wavelet-based RKHS, from which we can induce a scaling kernel. Suppose  $\phi(x)$  is a real continuous scaling function such that [10]

$$\phi(x) = O(|x|^{-1-\tau}) \text{ as } x \rightarrow \pm\infty, x \in R \tag{6}$$

$$\phi(\Omega) = \sum_n \phi(n) \exp(-2\pi i \Omega n) \neq 0 \quad \Omega \in R \tag{7}$$

and whose translates  $\{\phi(x-k)\}$  form an orthonormal basis of a subspace  $V_0$  of the square integrable space  $L^2(R)$ . We further suppose that an associated multiresolution analysis (MSA) of closed subsets of  $L^2(R)$ ,  $\{V_j\}_{j \in Z}$ , exists and satisfies [11]

$$\dots V_{-1} \subset V_0 \subset V_1 \subset \dots \subset V_j \subset \dots \subset L^2(R)$$

$$f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$$

$$\bigcup_{j \in Z} V_j = L^2(R), \quad \bigcap_{j \in Z} V_j = \langle 0 \rangle$$

Walter pointed that under weak hypotheses (6) and (7), each multiresolution subspaces  $\{V_j\}_{j \in \mathbb{Z}}$  is a RKHS [10].

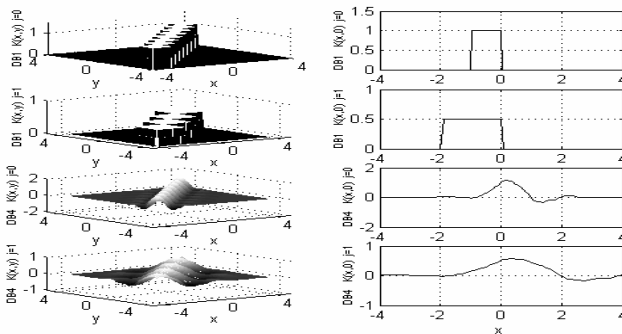
The famous Moore-Aronszajn theorem states that for every RKHS, there exists a unique reproducing kernel and vice versa [9]. The wavelet-based RKHS  $\{V_j\}_{j \in \mathbb{Z}}$  defined in the previous sections allows us to induce a corresponding scaling kernel

$$K_j(x, y) = \sum_k 2^{-j} \phi(2^{-j}x - k) \phi(2^{-j}y - k) \tag{8}$$

The formation of the scaling kernel (8) is a kernel of dot-product type. The Mercer theorem (see [12]) gives the conditions that a dot-product type kernel must satisfy. It is proved that the kernel (8) satisfies Mercer’s condition.

The scaling kernel (8) has a free parameter, the scale  $j$ , to be tuned. The problem of choosing the optimal values of  $j$  is called the model selection. A more disciplined approach is to use a validation set, or by data re-sampling techniques such as cross-validation and bootstrapping [13,14]. Alternatively, one can utilize an upper bound on the generalization error predicted by the theory of structure risk minimum (SRM). For simplicity, the optimal choice of the scale  $j$  has been solved by cross-validation in this paper.

The function  $\phi$  we chosen is the scaling function of the Daubechies (DB) wavelet with different orders. Suppose the interval of a compactly supporting  $\phi$  be  $[l, r]$ , it can be inferred that  $k \in S$  where  $S = [\max(2^{-j}x - r, 2^{-j}y - r), \min(2^{-j}x - l, 2^{-j}y - l)]$ . Fig. 1 shows at the same time kernel is zero when condition  $|x - y| > (r - l)/2^j$  is satisfied.



**Fig. 1.** Daubechies1 (DB1) and DB4 scaling kernels at different scale  $j$ . The subplots in column 1 are three dimensional plots and column 2 are slices.

## 4 Least Squares Support Vector Approximation with Scaling Kernels

If the LS-SVM employs Eq. (8) as its kernel, the notation has some differences with the LS-SVM that employs traditional kernels. To make the notation consistent with MSA, we seek the following approximation in the scale subspaces  $V_j$

$$f_j(x) = \sum_k c_k^{(j)} \phi_{jk}(x) + b^{(j)} \tag{9}$$

To recover  $f_j(x)$  the Eq. (2) can be rewritten as

$$\min_{w,b,e} \frac{1}{2} \|f_j\|_H^2 + \frac{C}{2} \sum_{n=1}^N \xi_n^2, \tag{10}$$

where  $\|\cdot\|_H$  denotes the norm defined on a RKHS.

From [16] we can infer that

$$\|f_j\|_H^2 = \sum_k c_k^{(j)2}. \tag{11}$$

After substituting Eq. (11) into (10), the optimization problem becomes

$$\min_{w,b,e} \frac{1}{2} \sum_k c_k^{(j)2} + \frac{C}{2} \sum_{n=1}^N \xi_n^2 \tag{12}$$

subject to the equality constraints

$$t_n - \mathbf{w}^T \boldsymbol{\varphi}(x_n) - b^{(j)} = \xi_n, \quad n = 1, 2, \dots, N$$

For the purpose of solving the above constrained optimization problem, the technique of lagrange multipliers is used, and the lagrangian corresponding to the above problem is defined

$$G = \frac{1}{2} \sum_k c_k^{(j)2} + \frac{C}{2} \sum_{n=1}^N \xi_n^2 + \sum_{n=1}^N \alpha_n^{(j)} (t_n - f_j(x_n) - \xi_n) \tag{13}$$

where  $\alpha_n^{(j)}$  are Lagrange multiplier at the scale  $j$ . The coefficient  $c_k^{(j)}$  can be obtained by setting the derivatives with respect to  $c_k^{(j)}$  zero, in final one arrives at

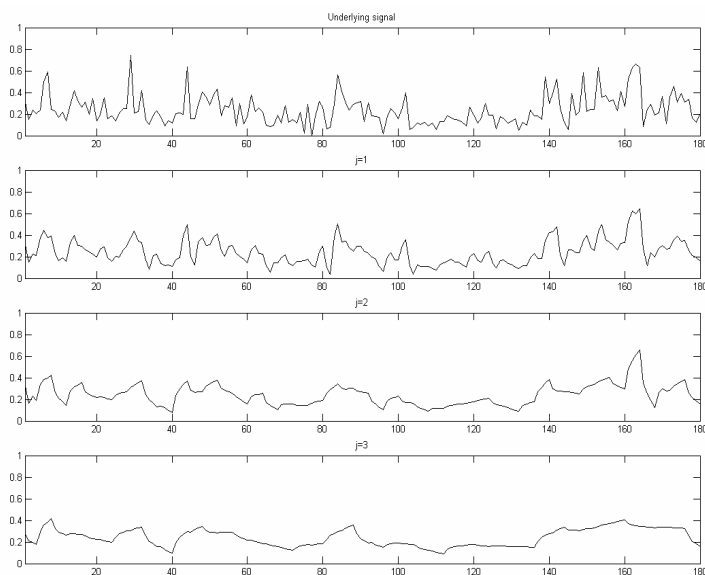
$$f_j(x) = \sum_{n=1}^N \alpha_n^{(j)} K_j(x, x_n) + b^{(j)} \tag{14}$$

where the kernel function  $K_j$  is given by Eq. (8).

## 5 Experiments

This experiment illustrates the multi-scale approximation ability of LS-SVM with scaling kernels. In the experiment, the function  $\phi$  we chosen is the scaling function of the Daubechies wavelet with order 2. The regularization factor is set for  $C = 10^4$ . The multi-scale approximation results are shown in Fig.2.





**Fig. 2.** The underlying signal and the approximations at different scale  $j$

As seen in the figure, the LS-SVM with scaling kernels can provide a hierarchically multi-scale approximation for the given target signal. At different scale, the approximation generally characterizes different physical structures of the signal.

## 6 Conclusions

Many of characteristics of least squares support vector machine (LS-SVM) are determined by the type of kernels used. Traditional kernels such as polynomial kernel and radial basis function kernel have many demerits. It is valuable to investigate the problem of whether a better performance could be obtained if we construct a scaling kernel by using the scaling function. This thesis presents a way for building a wavelet-based reproducing kernel Hilbert spaces (RKHS) and its associate scaling kernel for LS-SVM. Results on the approximation problems illustrate that the LS-SVM with scaling kernel can give better approximation performance and can implement multi-scale approximation.

## References

1. Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. *IEEE Trans Neural Networks* 12(2), 181–201 (2001)
2. Cristianini, N., Taylor, J.S.: *An Introduction to Support Vector machines*. Cambridge University Press, Cambridge (2000)
3. Suykens, J.A.K., Gestel, T.V., Brabanter, J.D., et al.: *Least Squares Support Vector Machines*. World Scientific Pub. Co., Singapore (2002)

4. Suykens, J.A.K., Vandewalle, J.: Recurrent least squares support vector machines. *IEEE Trans. Circuits and Syst. I.* 47, 1109–1114 (2000)
5. Zhang, L., Zhou, W.D., Jiao, L.C.: Wavelet support vector machine. *IEEE Trans. System, Man, Cybernetics* 34, 34–39 (2004)
6. Opfer, R.: Tight frame expansions of multiscale reproducing kernels in Sobolev spaces. Technical Report, Institut für Numerische und Angewandte Mathematik, Universität Göttingen (2004)
7. Amato, U., Antoniadis, A., Pensky, M.: Wavelet kernel penalized estimation for non-equispaced design regression. *Statistics and Computing* 16(1), 37–55 (2006)
8. Rakotomamonjy, A., Canu, S.: Frame, reproducing kernel, regularization and learning. *Journal of Machine Learning Research* 6, 1485–1515 (2005)
9. Aronszajn, N.: Theory of reproducing kernels. *Trans. Amer. Math. Soc.* 68, 337–404 (1950)
10. Walter, G.G.: A sample theorem for wavelet subspaces. *IEEE Trans. Information Theory* 38, 881–884 (1992)
11. Mallat, S.: A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. PAMI* 11, 674–693 (1989)
12. Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy Soc. London A209*, 415–446 (1909)
13. Chapelle, O., Vapnik, V.: Model selection for support vector machines. A. Solla. In: *NIPS. Advances in Neural Information Processing Systems*, vol. 12, pp. 349–355. MIT Press, Cambridge (2000)
14. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. Wiley, New York (2002)
15. Girosi, F.: An Equivalence between Sparse Approximation and Support Vector Machines. *Neural Computation* 10, 1455–1480 (1998)

# Evolutionary Feature and Parameter Selection in Support Vector Regression

Iván Mejía-Guevara<sup>1</sup> and Ángel Kuri-Morales<sup>2</sup>

<sup>1</sup> Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas (IIMAS),  
Universidad Nacional Autónoma de México (UNAM),  
Circuito Escolar S/N, CU, 04510 D. F., México  
`imejia@uxmcc2.iimas.unam.mx`

<sup>2</sup> Departamento de Computación, Instituto Tecnológico Autónomo de México,  
Río Hondo No. 1, 01000 D. F., México  
`akuri@itam.mx`

**Abstract.** A genetic approach is presented in this article to deal with two problems: a) feature selection and b) the determination of parameters in Support Vector Regression (SVR). We consider a kind of genetic algorithm (GA) in which the probabilities of mutation and crossover are determined in the evolutionary process. Some empirical experiments are made to measure the efficiency of this algorithm against two frequently used approaches.

**Keywords:** Support Vector Regression, Self-Adaptive Genetic Algorithm, Feature Selection.

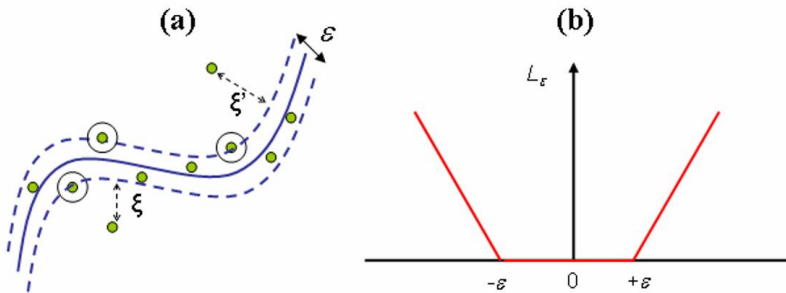
## 1 Introduction

Support Vector Machines (SVMs) have been extensively used as a classification and regression tool with a great deal of success in practical applications [1] [2] [3]. Some of the advantages of SVMs over other traditional methods (as Neural Networks) are: a) The development of sound theory first, then implementation and experimentation, b) The solution to an SVM is global and unique, c) They have a simple geometric interpretation and yield a sparse solution, d) The computational complexity of SVMs does not depend on the dimensionality of the input space, e) SVMs take advantage of structural risk minimization and f) They are less prone to overfitting [4] [5]. The calibration of parameters for a SVM is an important aspect to be considered since its learning and generalization capacity depends on their proper especification [6]. In this article, we focus on the problem of nonlinear regression and we propose a genetic approach for the optimization of those parameters. The method that we propose also considers the solution to the problem of feature selection and has the property to be self-adaptive since the probabilities of crossover and mutation are determined in the evolutionary process.

The article is organized as follows. We discuss in section 2 some theoretical characteristics of SVR. In section 3, we describe a self-adaptive Genetic Algorithm approach. In section 4 we describe some experiments where other approaches are used to solve the problem of determining the proper selection of parameters in SVR and the results obtained from comparisons performed against these approaches. Finally, some conclusions are presented.

## 2 Support Vector Regression

SVM is a supervised method discovered by Vapnik *et al* [7] and has been extensively used in the solution of pattern classification, nonlinear regression and other problems. In this section we focus on the problem of regression, where a set of data (the training dataset)  $\tau = \{x_i, y_i\}_{i=1}^N$ , is considered for the training process, where  $y_i, i = 1, \dots, N$  are continuous output values. Given this training set, the goal of SVR is to approximate a linear function of the form  $f(x) = \langle w, x \rangle + b$  with  $w \in R^N$  and  $b \in R$  that minimizes an empirical risk function defined by  $R_{emp} = \frac{1}{N} \sum_{i=1}^N L_\epsilon(\hat{y} - f(x))$ , where  $L_\epsilon(\hat{y} - f(x)) = |\xi| - \epsilon$ , if  $|\xi| > \epsilon$  and 0 in other case. The term  $\xi$  is called *slack variable* and is introduced to cope with otherwise infeasible constraints of the optimization problem [8]. In other words, errors are disregarded as long as they are smaller than a properly selected  $\epsilon$  as shown in Figure 1.a). The last function is called *epsilon-insensitive loss function*, but it is important to point out that it is possible to use other kinds of loss functions in SVR. In Figure 1.b) a graphical representation of it is presented. In order to estimate  $f(x)$  a quadratic problem must be solved, which



**Fig. 1.** (a) Nonlinear Regression Function and penalization of points beyond limits of the  $\epsilon$ -tube, (b)  $\epsilon$ -Insensitive Loss Function

has the objective to minimize the empirical risk function. The dual form of this optimization problem is more appropriate. This formulation is as follows:

$$Max_{\alpha, \alpha^*} -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(x_i, x_j)$$

$$\begin{aligned}
 & -\epsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \tag{1} \\
 \text{s.t. : } & \sum_{j=1}^N (\alpha_j - \alpha_j^*) = 0 \\
 & \alpha_i, \alpha_i^* \in [0, C]
 \end{aligned}$$

The regularization parameter  $C > 0$  determines the tradeoff between the flatness of  $f(x)$  and the allowed number of points with deviations larger than  $\epsilon$ . The value of  $\epsilon$  is inversely proportional to the number of support vectors (represented by  $(\alpha_i - \alpha_i^*) \neq 0$ ) [8]. An adequate determination of  $C$  and  $\epsilon$  is needed for a proper solution of the problem. Some approaches has been proposed in the past for their determination either for the C parameter [9] or for both [10]. The determination of these parameters is the main objective here, so the method we propose is explained in the next section.

Functional  $K(x_i, x_j)$  in (1) is known a kernel function and it allows to project the original problem to a higher dimensional feature space where the dataset has a high probability to be linearly separable. Many functions can be used as kernels, but only if they fulfill Mercer’s theorem [11]. Some of the most popular kernels discussed in the literature are the radial basis functions (RBF) (2) and the polynomial kernel (PK) (3). In this paper we focus on the latter and we used them to compare the accuracy of the algorithm that we suggest against the one obtained with other alternatives. The expressions that characterize these kernels are as follows:

$$K(\mathbf{x}, \mathbf{x}_i) = e^{-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}} \tag{2}$$

$$K(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x} \cdot \mathbf{x}_i)^\rho \tag{3}$$

The parameters  $\sigma$  and  $\rho$  have to be properly determined in order for the SVM to generalize well. From now on, we refer to these parameters as kernel pameters ( $kP$ ), when we use either polynomial or Gaussian kernels in the experiments we will describe below. The selection of these parameters is very important because the  $kP$  determines the complexity of the model and affects its accuracy. For that reason, different approaches have been proposed in the past for its optimal selection [12] [13]. Their determination is also a very important objective and, therefore, we propose an alternative genetic approach.

Once the solution of (1) is obtained, the support vectors are used to construct the following regression function:

$$f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(x, x_i) + b \tag{4}$$

### 3 Self-adaptive Genetic Algorithm

Genetic Algorithms were formally introduced in the 1970s by John Holland and his students at the University of Michigan. Their advantage over other

computational systems has made them attractive for some types of optimization. In particular, GAs work very well on mixed (continuous and discrete) combinatorial problems. They are less susceptible to getting stuck at local optima than gradient search methods [14].

To use GAs some issues must be taken into account to ensure their proper functionality: a) genome representation, b) fitness function, c) initial population, d) selection method, e) probabilities for crossover and mutation and f) termination criteria [15] [16].

We considered a population of size  $P = 30$ . The initial population was randomly generated. Weighted binary fixed point representation was used. With this representation the range of possible values for a real  $r$  is  $-2^I \leq r \leq +2^I$ , where  $I$  is the number of bits in the integer part of  $r$ . A fixed point format has been used because of its good performance in constrained optimization problems [17].

Once the initial population is generated, Vasconcelos' model is used. This model considers full elitism and deterministic coupling, as follows. The genome is considered to be a ring of size  $\ell$ . Individuals  $i$  and  $n - i + 1$  are deterministically selected. A random number is generated; if it is smaller than  $Pc$  (the probability of crossover) then a semi-ring of size  $\ell/2$  is taken from each of the two parents; the resulting genomes pass on to the next population. Otherwise, the individuals are passed to the next population untouched. Uniform mutations occur with probability  $Pm$  [18].

The self-adaptive mechanism used here consists on including the crossover and mutation parameters in the genome and leaving the problem of their determination to the GA. This principle is known as population self-determination, which is described and experimentally tested in [19] and it follows an individualist principle where each individual of the population is affected by the parameters. This mechanism is applied here as is described now:

- (i) Crossover Probability ( $P_c$ ): if  $p_c$  represents the crossover probability for an individual. Then, for each individual of the population we have a crossover probability expressed as  $(p_c)_i, i = 1, \dots, N_p$  and, therefore, the value for  $P_c$  in each generation is computed by the following expression:  $\sum_{i=1}^{N_p} (p_c)_i / N_p$ .
- (ii) Mutation Probability ( $P_m$ ): if  $p_m$  represents the mutation probability for an individual. Then, for each individual of the population we have a mutation probability expressed as  $(p_m)_i, i = 1, \dots, N_p$  and, therefore, the value for  $P_m$  in each generation is computed by the following expression:  $\sum_{i=1}^{N_p} (p_m)_i / N_p$ .

## 4 Genetic Support Vector Regression and Feature Selection

### 4.1 Genetic Support Vector Regression

We explained here the characteristics of the genetic approach we propose for the specification of parameters and variable selection in SVR. We called this

approximation as Genetic Support Vector Regression (GSVR) and its implementation is as follows:

- (1) Define the genome for the representation of the parameters:  $C$ ,  $kP$ ,  $\epsilon$ ,  $P_c$ ,  $P_m$  and  $fS$
- (2) Randomly generate the initial population,
- (3) Compute the fitness for each individual of the population,
- (4) Apply genetic operations based on Vasconcelo's method and the self-adaptive approach,
- (5) If a termination criterion is reached, finish. If not, return to step 3.

Similar approaches have been used for classification problems in [20] [21], but new characteristics are considered here that have proven their efficiency in the past [18]: a) The fixed-point codification, b) The use of an efficient genetic strategy (Vasconcelos) and, b) The self-adaptive mechanism where  $P_c$  and  $P_m$  are determined in the evolutionary process.

The parameter  $fS$  in the step (1) will be useful for the implementation of feature selection, but the details are explained in the next section. Concerning step (3), the fitness is defined as the Mean Square Error (MSE), computed after the application of Cross-Validation. Other alternative is the estimation of the MSE defined on a test dataset, which is a faster way to compute the error, but a statistical analysis of the application of this approach in classification problems showed that the variance is higher [12] and for that reason we decided to use the Cross-Validation error.

The MSE is computed for each individual in the population and for the new individual generated in the evolutionary process through the application of the Sequential Minimal Optimization (SMO) algorithm, which is an efficient and fast way to train SVMs for classification and regression problems. The algorithm for classification was discovered by Platt [22] and an efficient algorithm for regression training with SMO is due to [23]. The kind of algorithm that is chosen in this phase is very important since most of the training time depends on its selection. The use of traditional optimization methods here is out of the question, because of the time these algorithms could need for training. Therefore, although some other algorithms can be used for training the SVM, we strongly suggest the use of the SMO approach.

Finally, step (5) refers to the stopping criterion for the GA. The one that we used is based on the number of generations, where only 50 generations were needed to obtain a competitive machine. For instance, in [20] the stopping criteria was 600 generations or that the fitness value does not improve during the last 100 generations. Moreover, 500 individual were used in that application in comparison with the 30 individual we use here. Those differences imply a significant decrease on the computation cost as we show in our experiments.

## 4.2 Feature Selection

A problem that has been a matter of study for many years is the selection of a subset of relevant variables for building robust learning methods. The terminology

of variable selection, feature reduction, attribute selection or variable subset are equivalent ways used in the literature when referring to this problem. The objective of variable selection is three-fold: a) improving the prediction performance of a machine learning, b) providing faster and more cost-effective predictors, and c) providing a better understanding of the underlying process that generated the data [24]. The evolutionary procedure used here consists of introducing a binary string of size equal to the number of independent variables of  $\tau$ , where a '0' in position  $i$  means that the variable  $i$  must be dropped during the training process and a '1' means the opposite. For instance, if the problem in question has 7 variables, the string 0101011 means that only the variables 2, 4, 6 and 7 must be used for training.

Once all the variables that will be specified in the genome have been defined the size of the genome for the GSVR is equal to  $n_C + n_{kp} + n_\epsilon + n_{pc} + n_{pm} + m$ , where  $n_C$ ,  $n_{kp}$ ,  $n_\epsilon$ ,  $n_{pc}$ ,  $n_{pm}$  and  $m$  are the number of bits used for the codification of  $C$ ,  $kP$ ,  $\epsilon$ ,  $P_c$ ,  $P_m$  and the number of characteristics, respectively.

Given the elitism property of the Vasconcelos' GA, the problem of sacrificing fitness when some variables of the training set are not considered is avoided since the evolutionary process begins with all variables, but a reduced subset of those variables is always kept.

## 5 Experiments and Results

In this section we use the genetic approach explained before for the determination of parameters in SVMs applied in the solution of nonlinear regression problems. To prove the efficiency of this method we compare its results against other two approaches used in the past for similar purposes. The first method is due to Cherkassky (CHK) *et al* [10], who proposed an analytical calculation of  $C$  and  $\epsilon$ . The other approach consists of the use of Cross Validation (CV), which allows the calibration of  $C$ ,  $\epsilon$  and the kernel parameter.

The CHK method suggests the use of  $C = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|)$  for the estimation of  $C$ , where  $\bar{y}$  and  $\sigma_y$  are the mean and standard deviation of the output values of the training set, respectively. The last expression is applicable when a radial basis kernel function is used and it is derived taking into account the regression function (4) and the constraints in (1), where the support vectors and  $C$  are involved. The evaluation of  $\epsilon$  is reached computing  $\epsilon \propto \frac{\sigma}{\sqrt{n}}$ , where  $\sigma$  is the noise deviation of the problem. However, this methodology is only applicable for low values of  $N$ , since higher ones imply a very low  $\epsilon$ . Given this drawback, the author recommends the use of  $\epsilon = \tau\sigma\sqrt{\frac{\ln N}{N}}$  instead,  $\tau$  being a constant which must be empirically determined and  $\sigma$  is calculated in practice using the following equation:

$$\hat{\sigma} = \frac{1}{n-d} \sum_{i=1}^N (y - \hat{y})^2 \quad (5)$$

where  $\hat{y}$  is the estimated output value gotten from the application of high order polynomials<sup>1</sup>. We use Minimax Polynomial Approximation (MPA) to deal with

<sup>1</sup> The author also suggests other approach, but we use this one.



this problem. It is important to mention that with this technique it is not possible to estimate  $kP$  and this is done here by using some arbitrary values and applying Cross-Validation once the proper values of  $C$  and  $\epsilon$  had been estimated.

The Cross-Validation alternative is applied for the selection of parameters by choosing -arbitrarily or randomly- some values for each parameter. The proper values are chosen as the ones with the lowest CV error, where  $k$ -fold or Leave-one-Out (LoO) Cross Validation are very common in practice.

The datasets used for the comparison of the three methods were taken from the University of California at Irving Machine Learning Repository<sup>2</sup> (UMLR) which are labeled as mpg, mg and diabetes. The first dataset refers to the problem of predicting the efficiency in fuel consumption for several kinds of car models, it consists of 7 characteristics and 392 observations. The second problem has 1385 points with 6 independent variables each<sup>3</sup>. The third one concerns the study of the factors affecting patterns of insulin-dependent diabetes mellitus in children and it has 43 observations with 2 continuous characteristics<sup>3</sup>. We calculated the MSE for each experiment after applying 5-fold Cross-Validation. We also computed the execution time using a 1.8 MH Intel-Centrino processor with 512 RAM memory.

Two kernel functions were used during these experiments: a kernel function and a radial Gaussian basis function. As mentioned before, the parameters for these kernels were genetically calibrated for the GSVR approach and using 5-fold CV in every case.

MSEs, parameter values and execution times for each algorithm are shown in Table 1<sup>4</sup> (The label GSVR-RBF stands for *Genetic Support Vector Regression* trained with a RBF and so on.)

According with the results in Table 1, we can appreciate that GSVR approach is very competitive in comparison with the other methods. MSEs in the genetic method were at almost the same or lower than the ones obtained with CHK method, but the execution time was better. However, the most important advantage of GSVR is that it is of more general application because of its flexibility in the use of other kinds of kernel function in the training process. This is not possible with the CHK method.

Performance of the CV method is not conclusive in comparison with GSVR because it is worse than the one for diabetes, the same for mg and better for mpg for the two kinds of kernels, besides the performance of GSVR with a polynomial kernel is slightly better than the performance with RBF kernel. The problem with CV is the computation time, because it is significantly larger than the computation in GSVR for mpg and mg. The time for CV is almost 8 times larger than GSVR computation time in mpg with 30 individuals and 50 generations, using a RBF and almost twice using the polynomial kernel. In the case of

<sup>2</sup> <http://mllearn.ics.uci.edu/MLRepository.html>

<sup>3</sup> This data can be downloaded from:

<http://www.liacc.up.pt/ltorgo/Regression/DataSets.html>

<sup>4</sup> The time for CHK depends on the algorithm that is used for the estimation of  $\sigma$  and, for that reason, it is not reported in this Table. However, this time was similar or smaller than the one spent on the other approaches.

**Table 1.** MSE, time (in minutes) and parameters estimated with CHK method, CV method and GSVR, using RBF and PK

Problems	Time	MSE	C	$\epsilon$	kP
<b>mpg</b>					
GSVR-RBF	6.22	6.59	15.85	0.31	0.53
GSVR-PK	48.07	7.16	4.40	0.46	3
CV-RBF	46.64	6.81	13.34	0.84	0.5
CV-PK	92.71	7.17	43.49	0.84	3
CHK		7.17	46.86	0.47	0.5
<b>diabetes</b>					
GSVR-RBF	1.38	0.32	14.97	0.49	0.2145
GSVR-PK	1.32	0.31	1.23	0.49	2
CV-RBF	1.53	0.28	5.56	0.68	0.5
CV-PK	0.87	0.28	5.56	0.70	2
CHK		0.34	6.91	0.51	0.5
<b>mg</b>					
GSVR-RBF	28.23	0.01	2.44	0.09	3.6602
GSVR-PK	847.80	0.02	30.91	0.10	3
VC-RBF	120.02	0.01	0.09	0.06	1.1
VC-PK	939.00	0.02	8.77	0.01	3
CHK		0.02	1.61	0.05	0.8

diabetes, the time for both methods is almost the same, maybe because the number of observations for this dataset is the smallest (the difference is no greater than 0.5 minutes for the entire training and optimization). In mg, the computation time is 4 times larger for CV with a RBF and 50 minutes larger with the polynomial kernel. Other problem with CV is the selection of candidates as optimal parameters in the process of optimization. Even when this operation is done randomly, the range of the values for those parameters is not clear. Moreover, just a limited number of values for those parameters can be chosen with this technique. With the GSVR method, on the other hand, the problem of the range is similar but there are more possibilities.

Feature selection is other important characteristic of the GSVR and it is also another advantage since CHK method does not consider how to tackle this problem and CV method could be applied in [25] but with a significant additional computation cost, which actually is its main disadvantage as we mentioned before.

## 6 Conclusions

A new algorithm was presented in this article to tackle the problem of feature selection and the calibration of parameters in SVR. The proposed method, named GSVR, was superior in comparison with two approaches used in the past for several reasons: a) the fitness of GSVR was the same or better in the majority of cases, b) the computation time is significantly smaller than the time CV takes

for the same problems, c) the GSVR is most robust than Cherkassky approach since it can be applied in principle with many different kinds of kernel functions and, d) feature selection is implemented with GSVR with no additional computational cost and without loss in accuracy. A robust methodology for the statistical validation of different machine learning methods designed for tackling nonlinear regression problems is a matter of future work.

## References

1. Guyon, W., Barnhill, V.: Gene selection for cancer classification using support vector machines. *MACHLEARN: Machine Learning* 46 (2002)
2. Huang, Z., Chen, H., Hsu, C.J., Chen, W.H., Wu, S.: Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision Support Systems* 37(4), 543–558 (2004)
3. Kim, K.J.: Financial time series forecasting using support vector machines. *Neurocomputing* 55(1-2), 307–319 (2003)
4. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2(2), 121–167 (1998)
5. Cristianini, N., Shawe-Taylor, J.: *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge (2000)
6. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. *Machine Learning* 46(1/3), 131 (2002)
7. Boser, B.E., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: *COLT*, pp. 144–152 (1992)
8. Smola, A., Schölkopf, B.: A tutorial on support vector regression (2004)
9. Kuri-Morales, A., Mejía-Guevara, I.: Evolutionary training of svm for multiple category classification problems with self-adaptive parameters. In: Simão-Sichman, J., Coelho, H., Oliveira-Rezende, S. (eds.) *IBERAMIA/SBIA 2006*. LNCS (LNAI), pp. 329–338. Springer, Heidelberg (2006)
10. Cherkassky, V., Ma, Y.: Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks* 17(1), 113–126 (2004)
11. Haykin, S.: *Neural networks: A comprehensive foundation*. MacMillan, New York (1994)
12. Mejía-Guevara, I., Kuri-Morales, A.: Genetic support vector classification and minimax polynomial approximation (2007), <http://www.geocities.com/gauss75/ivan.html>
13. Friedrichs, F., Igel, C.: Evolutionary tuning of multiple svm parameters. *Neurocomputing* 64, 107–117 (2005)
14. Holland, J.H.: *Adaptation in natural artificial systems*. University of Michigan Press, Ann Arbor (1975)
15. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading (1989)
16. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge (1996)
17. Kuri, A.: *A Comprehensive Approach to Genetic Algorithms in Optimization, and Learning. Theory and Applications, Foundations*. vol. 1. IPN (1999)
18. Kuri-Morales, A.F.: A methodology for the statistical characterization of genetic algorithms. In: Coello Coello, C.A., de Albornoz, Á., Sucar, L.E., Battistutti, O.C. (eds.) *MICAI 2002*. LNCS (LNAI), vol. 2313, Springer, Heidelberg (2002)

19. Galaviz, J., Kuri, A.: A self-adaptive genetic algorithm for function optimization. In: ISAI/IFIPS, p. 156 (1996)
20. Huang, C.L., Wang, C.J.: A ga-based feature selection and parameters optimization for support vector machines. *Expert Syst. Appl.* 31(2), 231–240 (2006)
21. Min, S.H., Lee, J., Han, I.: Hybrid genetic algorithms and support vector machines for bankruptcy prediction. *Expert Syst. Appl.* 31(3), 652–660 (2006)
22. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.) *Advances in Kernel Methods — Support Vector Learning*, pp. 185–208. MIT Press, Cambridge (1999)
23. Flake, G.W., Lawrence, S.: Efficient svm regression training with smo. *Machine Learning* (2001)
24. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
25. Abe, S.: Modified backward feature selection by cross validation. In: *ESANN*, pp. 163–168 (2005)

# Learning Models of Relational MDPs Using Graph Kernels

Florian Halbritter and Peter Geibel

Institute of Cognitive Science, University of Osnabrück  
Albrechtstrasse 28, 49076 Osnabrück, Germany  
{fhalbrit,pgeibel}@uos.de

**Abstract.** Relational reinforcement learning is the application of reinforcement learning to structured state descriptions. Model-based methods learn a policy based on a known model that comprises a description of the actions and their effects as well as the reward function. If the model is initially unknown, one might learn the model first and then apply the model-based method (indirect reinforcement learning). In this paper, we propose a method for model-learning that is based on a combination of several SVMs using graph kernels. Indeterministic processes can be dealt with by combining the kernel approach with a clustering technique. We demonstrate the validity of the approach by a range of experiments on various *Blocksworld* scenarios.

## 1 Introduction

In the past decade, reinforcement learning (RL) has received a lot of attention in the field of artificial intelligence and machine learning [1,2]. More recently, various researchers have suggested an extension of this technique called *relational reinforcement learning* (RRL) [3,4,5,6]. In contrast to the traditional approach, where states are usually represented as some sort of feature vector, RRL makes use of structured representations. A relational view of states and actions allows to abstract from the very situation itself to its structural features and therefore to generalize from specific tasks to the important characteristics of the problem.

In this article, we present an indirect model-based framework, in which we employ a series of support vector machines (SVMs; cp. [7]) to learn a model, which can thereafter be used as a basis for model-based RL methods to learn policies including dynamic programming methods like relational variants of value iteration and policy iteration [2,8,9,10]. The support vector machines use a powerful product graph kernel [11,4] to deal with graph-based representations of the state-action space.

The main contribution of our work is the learning of the model using a combination of SVMs. This way, we can extend the expressiveness of classical STRIPS-like production rule, and, although we focus on qualitative models in this article, our approach can be extended for including real-valued information easily; indeterministic models can be obtained by using a clustering method in a previous step, for grouping possible outcomes of an action application.

In order to show the feasibility of our approach, we evaluate our framework through numerous experiments on planning tasks in the *Blocksworld* [12,13] and show its clear superiority with respect to the need for training data over a model-free variant of our algorithm. Note that our focus is on learning the model. We therefore employ a relatively simple, straightforward method for learning the policy given the already learned model. The method requires complete retraining the value function from time to time. Although this method works well, it is possible to combine the model-learning approach with other model-based learning techniques (e.g. [8,9,10]).

The remainder of this paper is structured as follows: In section 2, we begin with a brief review of the current state of model-learning in RRL. Afterwards, in the sections 3 and 4, we explain the proposed framework in detail, by showing how we use SVMs to learn a model of the environment and how we utilize these to train an RRL agent. Experiments in the *Blocksworld* will be the focus of section 4. We present numerous experiments performed in different settings of the *Blocksworld* and the results obtained. Finally, we conclude with an extensive discussion of the implications of our results and this study as a whole and propose some issues for future work (section 6).

## 2 Model-Learning for Relational MDPs

Model-free RRL techniques [3,11] do not presuppose a known model of the process to be controlled. Instead, learning an evaluation function for state-action pairs allows to derive an optimal policy without having learned the model comprising the state transition probabilities and the reward function. On the other hand, it is well-known that model-based methods like Value Iteration and Policy Iteration usually converge faster because they can update a value function for states based on *all* its successor states [2,8,9,10]. Moreover, it is possible to combine model-free algorithms with model-based planning methods, see [14]. The work of Croonenborghs et al. is directly related to our approach: it uses probabilistic first order decision trees that specify for single literals their probability of being true in the successor state, assuming conditional independence. In contrast, we use an expressive, kernel-based method which also allows to predict quantitative information. Although the kernel part as such is deterministic, a clustering of successor states for the same action yield an indeterministic model that assigns probabilities to successor states as a whole instead of to a single literal, thus better accounting for correlations.

Classical RL approaches treat the states as being either atomic or fixed-length vectors. For atomic states, model learning (or estimation of the Markov Decision Process (MDP)) consists of learning the reward function (or distribution) together with determining which successor states can occur with what probability when an action is carried out in a state. The problem of learning a relational MDP additionally requires learning suitable action descriptions. For deterministic planning problems, one usually considers STRIPS-rules consisting of preconditions and effects, mostly represented by three lists of literals.

In the scientific literature, the induction of first order rules has been considered relatively rarely [15,16,17,18], although it had first been addressed in the 1970s [19]. Finding the relevant preconditions is either based on specializing a most general (empty) precondition, or by computing generalizations of appropriate training examples (graphs or logical descriptions). The effects of an action can be obtained by comparing a state and its successor state.

While Benson [16] and Wang [17] also consider examples of unsuccessful rule applications (negative examples), Pasula et al. [15] and Gil [18] assume the presence of positive examples only. In the latter case, either additional criteria for finding appropriate rules or the generation of negative examples is required. Here, we assume that examples of unsuccessful actions applications are given as well or might be generated from examples of other rules.

In this work, instead of explicitly generating STRIPS-like rules, we specify the model by trained support vector machines operating on annotated graphs. This allows us to incorporate numerical information in a straightforward manner and we are not restricted by properties of the STRIPS language or its variants.

### 3 Model Learning Using Graph Kernels

A model for RL has the sole purpose to predict the effects of actions that an agent carries out, i.e. the resulting state, as well as the immediate reward the agent obtains. In this work, we use a combination of SVMs to produce these predictions. An approximate, yet efficient way for describing graphs representing the states, is the use of so-called label sequences [20]. In this paper, we employ the product graph kernel [11,4], which provides a powerful means for comparing graphs with respect to common label sequences within them. The computation of the graph kernel for  $g$  and  $g'$  is based on the adjacency matrix,  $A$ , of the so-called product graph whose node set consists of pairs of nodes  $(n, n')$  from the original graphs. The kernel is computed as

$$k(g, g') = \sum_{r=1}^R \sum_{n, m \in g, n', m' \in g'} A^r((n, n'), (m, m')),$$

where  $A^r$  denotes the  $r$ -th power of the adjacency matrix of the product graph.  $A^r((n, n'), (m, m'))$  corresponds to the number of common label sequences of length  $r$  from  $n$  to  $m$  in  $g$  and from  $n'$  to  $m'$  in  $g'$ .  $R$  is the maximum path length, which has to be selected by the user. Note that by allowing continuous values in the product graph, we can generalize the kernel for using quantitative information. Using it e.g. with support vector regression thus allows to represent complex real-valued functions based on structural and numerical information.

As an input, all SVMs receive a graph-representation of the current state plus some additional information depending on the type of the SVM, see below. Such a graph has one node for each entity occurring in the state. The label of a node is either empty or corresponds to the type of the entity. In contrast to the type, node properties like color or size will be encoded by a labelled loop

for the respective node, i.e. an edge. Edges exist also between nodes for which certain relations hold (the labels corresponding to the names of the relations). Additionally, it is important to indicate the action that is to be carried out from this state in order to let the SVMs know how the state will be affected. To do so, we introduce an *action marker* to the state graph, that is, an additional node with a label corresponding to the name of the action and edges pointing from the marker node to each of the entities involved in the action. Furthermore, we augment the labels of the affected nodes to show that they are arguments of the action.

An example of a graph corresponding to a state in the Blockworld, as we will encounter later, can be seen in Fig. 1 (left). Each block is represented by a node, with the node type given in the respective circle (we left out the node identifiers from  $N$ ). Nodes are characterized by the two properties `onTable` and `clear` (i.e. there is no other block on top). `on` is the only (real) binary relation. The action of moving one block on top of another has been named `move` and is represented by a dashed circle. The arguments nodes of the action were labelled as `move/1` and `move/2`, respectively. The `move`-action node is connected to its arguments by edges also labeled `move/1` and `move/2`.

By adding the action to the graph, additional paths are introduced to the graph. These paths will be identical in all graphs with the same marker. Thus, the graph kernel will recognize a higher degree of similarity for graphs containing the same marker, because it is based on the number of common label sequences in the two graphs it operates on. The chosen action representation allows the model to consider actions affecting more than two nodes.

In order to determine the successor state when applying an action, we employ one SVM  $S_r$  for each type of feature/relation  $r$  in the state descriptions (examples for such features are `on`, `clear` and `onTable`). A single SVM  $S_r$  represents the effects of an action for a specific feature and operates on pairs of nodes (excluding action and query nodes, see below). In this work, we chose to predict if the respective label changes or not, although it also possible to predict edge/no edge directly, or a real value in the case of qualitative information like `distance`, etc. For each relation  $r$  the respective SVM will, hence, be called several times for each possible combination of two nodes in the graph<sup>1</sup> representing the state. For the respective pair, the SVM  $S_r$  predicts whether the corresponding relation will exist in the next state or not. In order to tell the SVM which pair of nodes we are considering at the moment, an additional marker, the *query marker*  $q(n_1, n_2)$ , is added in the same way as the action marker.

Combining the predictions of all SVMs  $S_r$  for all node pairs allows us to create the next state graph. In order to reduce the complexity (number of pairs), we assume that nodes affected by an action must be explicitly “mentioned” in the arguments of an action  $a(n_0, \dots, n_k)$ , which is a common assumption in AI problem solving, so not all pairs of nodes have to be considered.

<sup>1</sup>  $n$ -ary relations have not been investigated in this work. However, this would not pose a problem, in principle, since it is a well-known fact that arbitrary  $n$ -ary predicates can be decomposed into a set of binary predicates.



The model uses an additional SVM  $S_{\text{PRE}}$  solely for predicting whether an action is *applicable* in a state or not. This SVM, which operates on the state graph augmented with the action marker, checks the pre-conditions of the action. Since non-applicable actions leave the state unchanged, we can therefore neglect a majority of possibly costly transition predictions for non-applicable actions in favour of speeding up the whole process.

The model now consists of an SVM  $S_r$  for every relation  $r$ , which maps a state graph, an action marker, and a query marker to  $\{\text{change, nochange}\}$ . The successor state is obtained by combining the predictions for every relation and every node pair, given the current state. The SVM  $S_{\text{PRE}}$  operating on states with an action marker predicts the applicability of an action. A regression SVM  $S_{\text{REW}}$  determines the reward, when an action is applied to a state.

*Indeterministic processes* can be handled by first clustering the state transitions belonging to the same action  $m$ . This clustering of graphs can be achieved with standard techniques operating on a suitable measure of graph similarity or distance. We can, for instance, use the kernel as the similarity measure or we might resort to alternatives that have already been applied to such tasks successfully. The method described above is then applied to each single cluster. The proportion of the examples in each cluster can be used for estimating the probability for the corresponding state transition.

## 4 Model-Based Approximate, Asynchronous Value Iteration

Grounding on the SVM model just explained, any well-established model-based RL method can be used, in principle, to derive an optimal policy. For the purpose of this work, we decided to stick to use a variant of the Value Iteration [12] procedure which is a simple, yet powerful method for learning optimal policies in the case of discounted cumulative returns. Value Iteration is based on the update

$$V^{t+1}(x) = \max_u \left[ \sum_{x'} p_{x,u}(x') (r_{x,u}(x') + \gamma V^t(x')) \right]$$

where  $p_{x,u}(x')$  is the probability of reaching  $x'$  when action  $u$  is used in  $x$ , and  $r_{x,u}(x')$  is the respective reward.  $V^t$  denotes the  $t$ -th approximation of the optimal value function. The learnt model is necessary for evaluating the RHS of the equation.

Unfortunately, a pure table-based representation of the value function will be practically infeasible for all but the smallest state spaces. Therefore we again employ another support vector machine  $S_{\text{VAL}}$  acting on state graphs to store the value function. In the following, we will concentrate on typical planning problems where the task is to reach a goal state as quickly as possible. Since the reward for every single step is  $-1$ , we did not learn the reward function (i.e.,  $S_{\text{REW}}$ ), but assumed it to be known.

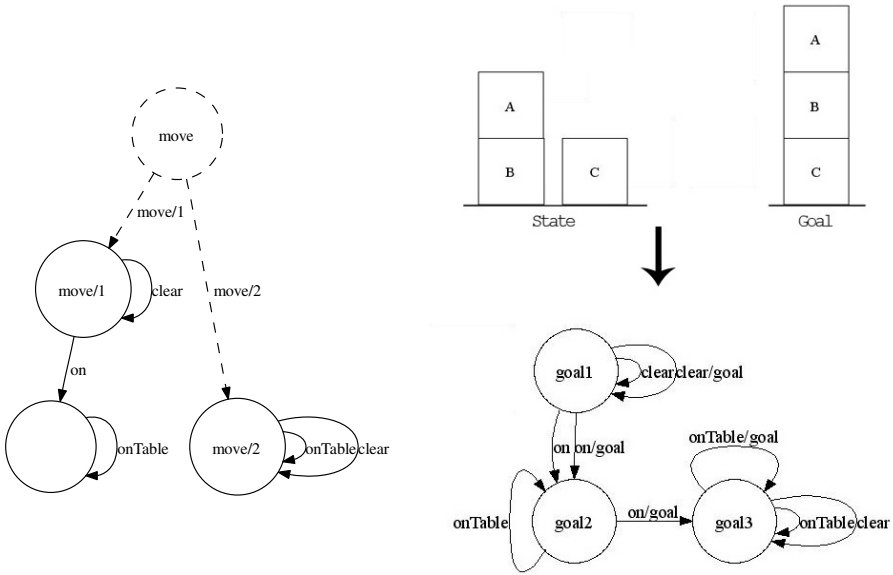


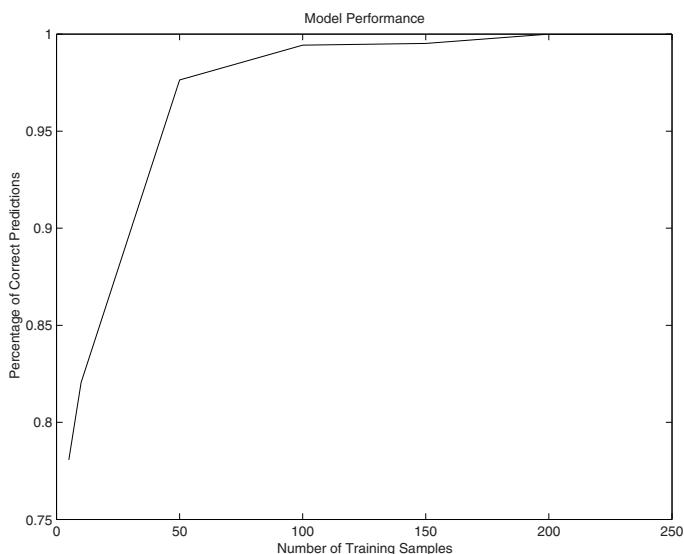
Fig. 1. *Blocksworld* states and graphs. For explanation see text.

Because the utility of a state does not ground purely on the state itself, but actually on the relation of the state to the goal state, it increases the flexibility to mark the goal conditions in the graph in order to predict the expected cumulative return with  $S_{VAL}$ . This can be easily accomplished by adding all relations from the description of the goal state to the original graph. In order to distinguish them from the normal relations in the state, their labels are augmented by adding a distinct string “goal” to them. Additionally, the labels of all nodes corresponding to the entities occurring in the goal conditions are changed by adding “goal” to them, too. Both an example state and the corresponding graph, are shown in Fig. 1 (right).

Value updates are performed asynchronously based on the current estimates. Since support vector machines are incapable of online learning, the function  $S_{VAL}$  needs to be retrained from scratch periodically, that is, whenever the number of erroneous predictions exceeds a certain threshold (the default value is 1, meaning updates are performed whenever a value estimate changes).

## 5 Experiments and Results

In this section we evaluate the proposed framework for model-based reinforcement learning by investigating its performance on several planning tasks in the *Blocksworld*. For all experiments, we used an extended version of the LIBSVM [21]. We will assess the performance of the model on predicting state transitions and the agent’s capability of learning optimal policies for stacking blocks in a particular order.



**Fig. 2.** Performance of the model: The graph shows the number of correct state predictions depending on the number of samples the model has been trained with

## 5.1 Problem Settings

A configuration in the *Blocksworld* consists of a pre-defined number of blocks and a table. Our aim is to stack the blocks in a specific order using a set of available actions. Only two actions are available: Moving a block  $X$  on top of another block  $Y$  ( $\text{move}(X, Y)$ ) and moving a block  $X$  onto the table ( $\text{moveTable}(X)$ ). The states themselves are characterized via various properties (features, predicates), here: A block  $X$  is on top of another block  $Y$  ( $\text{on}(X, Y)$ ), a block  $X$  is on the table ( $\text{onTable}(X)$ ) or a block  $X$  is clear ( $\text{clear}(X)$ ), i.e. there is no other block on top of it. Clearly, the actions only succeed if specific conditions hold. For example, we can only move block  $X$  onto block  $Y$  ( $\text{move}(X, Y)$ ), if both blocks are clear ( $\text{clear}(X), \text{clear}(Y)$ ). Evidently, a block  $Y$  cannot be clear, if there is another block  $Z$  on top of it ( $\text{on}(Z, Y)$ ). Thus, the purpose of our model is to learn which conditions apply to which actions and how the actions affect the states.

Representing the *Blocksworld* states as graphs can be done in a straightforward manner by introducing one node for each block mentioned in the state description and edges for the relations. Figure 1 shows an example of such a *Blocksworld* state with additionally an action having been marked in it in the way described in the previous chapter.

## 5.2 Results

In a first run of experiments we evaluated the capability of the proposed model to predict state transitions in *Blocksworlds* with three to eight blocks. We trained

**Table 1.** Percentage of correct predictions of transitions when confronting a perfect predictor trained in a simple environment with a more complex *Blocksworld*. The values in the brackets refer to the percentage of correct predictions when looking only at the positive samples, i.e. those where the action changed the state.

Trained with...	Tested with...	Accuracy
3 Blocks	5 Blocks	100.0 (100.0)
3 Blocks	8 Blocks	88.6 (72.6)
5 Blocks	8 Blocks	83.6 (32.4)
3 to 5 Blocks	8 Blocks	100.0 (100.0)

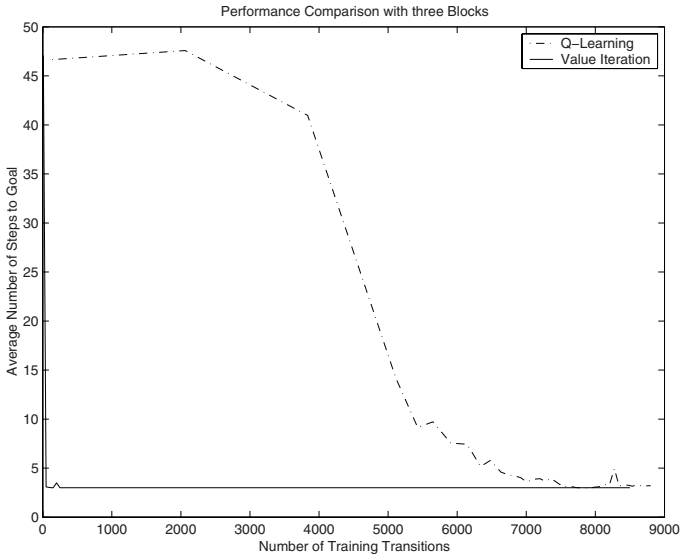
the model with increasingly many random transition samples (10, 50, 100, 150, 200, 250) once for each number of blocks. Afterwards we tested the model with 1000 further random transition examples and recorded the percentage of correct predictions. In order to reduce the effect of randomness, we repeated this procedure five times and took an average over the results. Figure 2 shows the results obtained.

Evidently, the model achieved a perfect prediction in all cases when provided with about 200 training samples. Note that for the easier cases drastically less samples were sufficient. In particular, it was possible to train a perfect model for three blocks with only 10 carefully hand-picked, most representative transition examples. It is furthermore interesting to notice that the mispredictions mostly resulted from ‘positive’ transitions, i.e. transitions where the state actually changed. The model can easily distinguish between successful and unsuccessful applications of actions, but needs some more training data to correctly predict state changes.

It is now important to assess the model’s generalization ability. To do so, we first trained the model with 250 transition samples with only three blocks. As we have seen previously, this is likely to give us a good predictor. Then we tested the performance on 1000 random transitions in a *Blocksworld* with 5 or 8 blocks and recorded the performance. Again, we averaged over five runs. Refer to Table 1 for the results.

The results prove that the model was able to generalize well to a slightly more complex environment, but was significantly flawed the more different the environment was from the one it was trained in. While it was still mostly able to distinguish successful from unsuccessful actions, predicting novel states posed a harder problem to the model. However, when we trained the model with 250 transitions samples with 3 to 5 blocks, a very good prediction could be achieved even in the 8-blocks scenario. In this case, the model apparently succeeded to extract the relevant characteristics of the problem and was no longer restricted to the one particular scenario it was trained with.

In a second batch of experiments, the RRL agent’s ability to find optimal policies on the basis of this model was to be investigated. In a similar fashion as before, we tested the agent in a 4-blocks *Blocksworld*, by first training its model with a varying number of transition samples and then allowing it to establish a policy using Value Iteration. After the training had converged, the



**Fig. 3.** Performance of the RRL agent in comparison to  $Q(\lambda)$ -Learning : The graph shows the average number of steps required to reach the goal step after the agent has been trained with a total of the stated transition samples

performance was tested using 100 random starting states and we recorded the number of steps it took to reach the goal (stacking all three blocks in a fixed order) in each run. If the agent failed to reach the goal within at most 50 steps, the run was considered to have failed, and a count of 50 steps was recorded. Again, we averaged over five runs. As a means for comparison, we furthermore implemented a  $Q(\lambda)$ -learning agent using replacing eligibility traces<sup>2</sup> and trained it to find an optimal policy. For this agent, we paused the learning procedure every 10 episodes and performed 100 test runs to assess its current performance. We also recorded the total number of actual transitions the agent had seen by that time in order to gain a comparable scale for the RRL agent. Figure 3 depicts the results obtained.

$Q(\lambda)$ -learning needed about 7000 transition examples (about 300 episodes) to find an optimal policy reaching the goal in some 3-5 steps (depending on the starting position). In contrast, the RRL agent could derive an equivalent policy from the model which needed only some 50 transition examples. The advantage could not be any more striking. Once the number of transition samples sufficed to create a perfect model, the agent could also learn an optimal policy. When provided with a flawed model, the policy will be largely random and therefore, least surprisingly, suboptimal.

<sup>2</sup> In a number of previous experiments we investigated the influence of the different parameters on the agent and eventually fixed them to the best possible choice, which was  $\alpha = 0.8$  (learning rate),  $\epsilon = 0.1$  (exploration rate) and  $\lambda = 0.6$  (influence of future experience on earlier state updates).

Additionally, we investigated the possibility of speeding up training by previously training the agent on a simpler task with only three blocks. A remarkable decrease in the number of steps required for Value Iteration to reach convergence could be observed: While usually about 1200 cycles were required before Value Iteration converged to an optimal policy, starting with an optimal policy found in the smaller *Blocksworld* (which usually needed some 100 cycles), convergence could be reached in only 375 cycles. Large parts of the previously learnt knowledge could apparently be reused for the more complicated scenario.

## 6 Conclusion

We have presented a framework for the automated creation of a powerful model for reinforcement learning and a simple learning procedure based on Value Iteration. An agent equipped with these tools has been shown to clearly outperform an agent based on  $Q(\lambda)$ -Learning with respect to the number of training data required to find an optimal policy. Furthermore, we were able to show that the agent was capable of benefiting from experience gained in a simpler, but fundamentally similar task when trying to learn a more complex scenario.

When comparing the results of our approach to those reported for the RRL-KBR procedure suggested by Gärtner and Driessens [4], we see that in general at least 100 episodes were required to find an optimal policy for their tasks, which amounts to thousands of transition examples. Other agents based on episodic, trial-and-error learning perform similarly. Moreover, they reported stacking blocks in a fixed order, in particular, to be a hard task for the agent to solve, which, however, did not pose any problems to our agent using the graph formalism proposed in this work. Future work, however, has to incorporate the comparison with model-based or indirect RRL learning methods like, for instance, FOALP, FLUCAP, ReBel, etc.

There is a lot of potential for future work in this field. Further investigations of the applicability of this framework to real-world tasks would be desirable. In a next step, we will also try to integrate the possibility of using real-valued properties (and possibly actions) into the graph kernel and the framework on the whole. First experiments with this have provided promising results, however, need further fine-tuning. Moreover, it will certainly be necessary to further improve the reinforcement learning component of the framework. While the Value Iteration procedure with the SVM-based function approximator was sufficient for the investigations at hand, it is unlikely that it will be computationally feasible for more complex tasks involving large graphs.

## References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
2. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-Dynamic Programming. Athena Scientific (1996)

3. Dzeroski, S., Raedt, L.D., Driessens, K.: Relational reinforcement learning. *Machine Learning* 43(1-2), 7–52 (2001)
4. Driessens, K., Ramon, J., Gärtner, T.: Graph kernels and gaussian processes for relational reinforcement learning. *Machine Learning* 64(1-3), 91–119 (2006)
5. Tadepalli, P., Givan, R., Driessen, K.: Relational reinforcement learning: An overview. In: *Proceedings of the ICML 2004 Workshop on Relational Reinforcement Learning* (2004)
6. van Otterlo, M.: A survey of reinforcement learning in relational domains. Technical report, CTIT Technical Report, TR-CTIT-05-31, July 2005, p. 70, CTIT Technical Report Series, ISSN 1381-3625 (2005)
7. Vapnik, V.N.: *The nature of statistical learning theory*. Springer, New York (1995)
8. Kersting, K., Otterlo, M.V., Raedt, L.D.: Bellman goes relational. In: Brodley, C.E. (ed.) *ICML, ACM, New York* (2004)
9. Scanner, S., Boutilier, C.: Approximate linear programming for first-order mdps. In: *Proceedings UAI 2005* (2005)
10. Hoelldobler, S., Karabaev, E., Skvortsova, O.: FluCaP: a heuristic search planner for first-order mdps. *JAIR* 27, 419–439 (2006)
11. Gärtner, T.: A survey of kernels for structured data. *SIGKDD Explor. Newsl.* 5(1), 49–58 (2003)
12. Russell, S.J., Norvig, P.: *Artificial intelligence: a modern approach*. Prentice-Hall, USA (1995)
13. Gupta, N., Nau, D.S.: Complexity results for blocks-world planning. In: *AAAI 1991. Proceedings of the Ninth National Conference on Artificial Intelligence*, vol. 2, pp. 629–633. AAAI Press/MIT Press, Anaheim, California, USA (1991)
14. Croonenborghs, T., Ramon, J., Blockeel, H., Bruynooghe, M.: Online learning and exploiting relational models in reinforcement learning. In: Veloso, M.M. (ed.) *IJCAI*, pp. 726–731 (2007)
15. Pasula, H., Zettlemoyer, L.S., Kaelbling, L.P.: Learning probabilistic relational planning rules. In: *ICAPS*, pp. 73–82 (2004)
16. Benson, S.: Inductive learning of reactive action models. In: *International Conference on Machine Learning*, pp. 47–54 (1995)
17. Wang, X.: Learning planning operators by observation and practice. In: *Artificial Intelligence Planning Systems*, pp. 335–340 (1994)
18. Gil, Y.: Learning by experimentation: Incremental refinement of incomplete planning domains. In: *ICML*, pp. 87–95 (1994)
19. Vere, S.A.: Inductive learning of relational productions. In: Waterman, D., Hayes-Roth, F. (eds.) *Pattern-Directed Inference Systems*, Academic Press, London (1978)
20. Geibel, P., Wysotzki, F.: Learning relational concepts with decision trees. In: Saitta, L. (ed.) *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 166–174. Morgan Kaufmann, San Francisco (1996)
21. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

# Weighted Instance-Based Learning Using Representative Intervals

Octavio Gómez, Eduardo F. Morales, and Jesús A. González

National Institute of Astrophysics, Optics and Electronics,  
Computer Science Department,  
Luis Enrique Erro 1, 72840 Tonantzintla, México  
{gomez, emorales, jagonzalez}@ccc.inaoep.mx  
<http://ccc.inaoep.mx>

**Abstract.** Instance-based learning algorithms are widely used due to their capacity to approximate complex target functions; however, the performance of this kind of algorithms degrades significantly in the presence of irrelevant features. This paper introduces a new noise tolerant instance-based learning algorithm, called WIB- $K$ , that uses one or more weights, per feature per class, to classify integer-valued databases. A set of intervals that represent the rank of values of all the features is automatically created for each class, and the nonrepresentative intervals are discarded. The remaining intervals (representative intervals) of each feature are compared against the representative intervals of the same feature in the other classes to assign a weight. The weight represents the discriminative power of the interval, and is used in the similarity function to improve the classification accuracy. The algorithm was tested on several datasets, and compared against other representative machine learning algorithms showing very competitive results.

**Keywords:** Feature Weighting, Instance-Based Learning,  $K$ -NN.

## 1 Introduction

Instance-based learning algorithms are derived from the nearest neighbor pattern classifier [6], and their design was also inspired by exemplar-based models of categorization [17]. Unlike other learning methods that construct an explicit description of the target function from the training examples, instance-based learning algorithms only store the examples, and delay the processing effort until a new instance need to be classified. Instance-based learning algorithms are widely used due to their advantages that include small training cost, efficiency gain through solution reuse [1], high capacity to model complex target functions and their ability to describe probabilistic concepts [2]. The performance of this kind of algorithms, however, degrades significantly in the presence of irrelevant features; so, distinguishing relevant features is a very important issue.

One way to improve the robustness of instance-based learning algorithms against irrelevant features is through feature weighting. In feature weighting,



each feature is multiplied by a weight value proportional to its ability to distinguish among classes. There are many algorithms of feature weighting. M. Tahir et al. (2007) [18] proposed a hybrid approach to simultaneously perform feature selection and feature weighting based on tabu search (TS) and the  $K$ -NN algorithms; they modified the solution encoding used by the TS algorithm by adding feature weights and binary feature vectors, and then used a  $K$ -NN classifier to evaluate the sets of weights produced by tabu search. Blansch et al. (2006) [4] proposed a method that performs a modular grouping of complex data called MACLAW. This method assigns weights to the features under a wrapper approach. A set of extractors is defined, and all the extractors are associated to a clustering algorithm and a local weights vector. The weights are obtained from standard cluster quality measures such as compactness. De la Torre et al. (2002) [19] applied the discriminative feature extraction (DFE) method to weight the contribution of each component of a feature vector. The weights are obtained from the partial probability weighting (PPW) exponents, and each weight represents the partial probability of each component of the feature vector. Thomas Gartner and Peter A. Flach (2000) [10] proposed an algorithm that combines naive bayes classification with feature weighting. They employ a support vector machine to weight the features, and, the weights are optimized to reduce the danger of overfitting. K. Kira and L. Rendell (1992) [12] proposed a weighting algorithm called RELIEF that estimates a feature weight  $W[A]$  as an approximation of the difference of probabilities  $P(\text{different value of } A \mid \text{nearest instance of different class}) - P(\text{different value of } A \mid \text{nearest instance of the same class})$  where  $A$  is an attribute. This algorithm was designed for 2-class problems. I. Kononenko (1994) [13] extended the RELIEF algorithm to deal with multi class problems, finding the probabilities with respect to each class and averaging their contribution. A good review and empirical evaluation of many feature weighting methods can be found in (Wettschereck et al., 1997) [21].

This paper introduces a new instance-based learning algorithm, called WIB- $K$ , that uses one or more weights, per feature per class, for the classification of integer-valued databases, and that is noise tolerant. A set of intervals that represents the rank of values of all the features is automatically created for each class, the representative intervals are located by means of a majority criterion, and the nonrepresentative intervals are considered as noise (outliers). The representative intervals of each feature are compared against the representative intervals of the same feature in the other classes to obtain a weight; this weight represents the discriminative power of the interval, and is used in the similarity function to improve the classification rate. The proposed algorithm was tested on several integer-valued datasets from the UCI repository, and compared against other representative machine learning algorithms, showing very competitive results.

The paper is organized as follows. Section 2 gives an overview of instance-based learning. In Section 3 the weighting schema is described. In Section 4 the experimental results are presented and, in Section 5, the main conclusions and a brief discussion of future work is given.

## 2 Instance-Based Learning

### 2.1 Learning Task and Framework

Instance-based learning algorithms are derived from the nearest neighbor pattern classifier [6]. This kind of algorithms stores and uses only selected instances to generate classification predictions by means of a distance function. The learning task of these algorithms is supervised learning from examples.

Each instance is represented by a set of attribute-value pairs, and all instances are described by the same set of  $n$  attributes. This set of  $n$  attributes defines an  $n$ -dimensional instance space. One of the attributes must be the category attribute and the other attributes are predictor attributes.

The primary output of an instance-based learning algorithm is a function, that maps instances to categories, called concept description; this concept description includes a set of stored instances and, possibly, information about the classifier past performance. The set of stored instances can be modified after each training instance is processed. All instance-based learning algorithms are described by the following three characteristics:

1. *Similarity function*: computes the similarity between a training instance  $i$  and the instances stored in the concept description. The similarities are numerical-valued.
2. *Classification function*: This function receives the results of the similarity function and the performance records stored in the concept description. It yields to a classification for the training instance  $i$ .
3. *Concept description updater*: Keeps the records of classification performance and decides the instances to be included in the concept description. It yields to a modified concept description.

The similarity and classification functions determine how the instances stored in the concept description are used to predict the category of the training instance  $i$ .

### 2.2 IB- $K$ Algorithm

IB- $K$  is a very straightforward instance-based learning algorithm. The distance function that it uses is:

$$Distance(x, y) = \sqrt{\sum_{i=1}^n f(x_i, y_i)}$$

where  $x$  is a test instance,  $y$  is a training instance,  $x_i$  is the value of the  $i$ -th attribute of instance  $x$  and  $f(x, y)$  is defined as follows:

$$f(x_i, y_i) = (x_i - y_i)^2$$

The instances are described by  $n$  features. The IB- $K$  algorithm is presented in Table [1](#).

**Table 1.** IB- $K$  algorithm ( $CD =$  concept description)

---



---

$CD \leftarrow$  all the labeled instances

**For each**  $x \in$  Training Set **do**

1. **For each**  $y \in CD$  **do**
  - $Dist \leftarrow Distance(x, y)$
  - If**  $Dist$  is one of the  $K$ -smallest distances  $Ksmall[m] \leftarrow Dist$
2.  $class(x) =$  majority class present in  $Ksmall[m]$
3.  $CD \leftarrow CD \cup x$

---



---

In order to label an instance, the IB- $K$  algorithm computes the distance between the test instance and the instances stored in the concept description, and stores the  $K$  nearest instances. The class of the test instance will be the preponderant class of the  $K$  nearest instances previously obtained.

### 3 Feature Weighting Based on Representative Intervals

#### 3.1 Initial Definitions

- $\Omega$  is the instance space formed by  $n$  instances and  $m$  features.
- $x_i \in \Omega$  represents the  $i$ -th instance,  $1 \leq i \leq n$ .
- $x_{i,j}$  represents the value of the  $j$ -th feature of the  $i$ -th instance,  $1 \leq j \leq m$ .
- $C_\beta^\alpha$  is a multiset that contains all the values of feature  $\beta$  for all the instances  $x_i \in \Omega$  with  $class(x_i) = \alpha$ .

$$C_\beta^\alpha = \{x_{i,j} | class(x_i) = \alpha \wedge j = \beta\}$$

- $D_\beta^\alpha$  is the set that contains all the values contained in  $C_\beta^\alpha$ , but without repeated values. This set is partially ordered under the function  $<$ . For example, if  $C_\beta^\alpha = \{3, 5, 7, 4, 9, 3, 9, 5, 3, 5, 4, 6\}$  then  $D_\beta^\alpha = \{3, 4, 5, 6, 7, 9\}$ .
- $f(a)$  is the frequency function, it returns the number of times that a value  $a \in D_\beta^\alpha$  appears in  $C_\beta^\alpha$ , and is defined as follows

$$f(a) = \sum_{\forall b_l \in C_\beta^\alpha} g(a, b_l)$$

where  $1 \leq l \leq |C_\beta^\alpha|$  and  $g(a, b_l)$  is defined as

$$g(a, b_l) = \begin{cases} 1 & \text{if } a = b_l \\ 0 & \text{otherwise} \end{cases}$$

For example, with the previous sets  $C_\beta^\alpha$  and  $D_\beta^\alpha$ ,  $f(5) = 3$ . This function can be viewed as histogram of the image.

### 3.2 Representative Intervals and Weights

First, the  $D_\beta^\alpha$  set must be partitioned into collectively exhaustive and mutually exclusive subsets  $D_{\beta,\gamma}^\alpha$ , where  $\gamma$  is the index of the partition. All the consecutive intervals must be grouped in exactly one partition. For example, if  $D_\beta^\alpha = \{1, 2, 3, 5, 6, 7\}$  then the only resultant partitions are  $D_{\beta,1}^\alpha = \{1, 2, 3\}$  and  $D_{\beta,2}^\alpha = \{5, 6, 7\}$ .

The magnitude of a partition  $D_{\beta,\gamma}^\alpha$  is:

$$\text{Magnitude}(D_{\beta,\gamma}^\alpha) = \sum_{\forall t \in D_{\beta,\gamma}^\alpha} f(t)$$

The amplitude of a partition  $D_{\beta,\gamma}^\alpha$  is:

$$\text{Amplitude}(D_{\beta,\gamma}^\alpha) = |D_{\beta,\gamma}^\alpha|$$

All the partitions  $D_{\beta,\gamma}^\alpha$  are grouped according to their amplitude. Let  $E_{\beta,\eta}^\alpha$  be the set formed by all the partitions  $D_{\beta,\gamma}^\alpha$  with the same amplitude  $\eta$ , then, the maximum amplitude  $\psi$  of the set  $E_{\beta,\eta}^\alpha$  is:

$$\psi = \text{argmax}(\text{Magnitude}(D_{\beta,\gamma}^\alpha))$$

where  $D_{\beta,\gamma}^\alpha \in E_{\beta,\eta}^\alpha$ . In order to discard the nonrepresentative intervals, considered as noise (outliers), it is necessary to define levels of confidence. This characteristic allows the algorithm to be noise-tolerant.

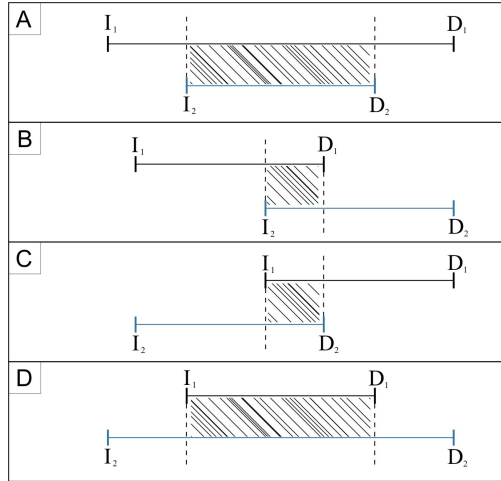
If  $\psi$  is the maximum amplitude of  $E_{\beta,\eta}^\alpha$  then the levels of confidence are shown in Table 2, where % represents the integer division.

**Table 2.** The four levels of confidence defined to discriminate noise

Level of confidence	Interval	Left value	Right value
High	$[H_i, H_f]$	$H_i = (H_f \% 2) + 1$	$H_f = \psi$
Medium	$[M_i, M_f]$	$M_i = (M_f \% 2) + 1$	$H_i - 1$
Low	$[L_i, L_f]$	$L_i = (L_f \% 2) + 1$	$M_i - 1$
Null	$[0, N_f]$	0	$L_i - 1$

The sets  $D_{\beta,\gamma}^\alpha \in E_{\beta,\eta}^\alpha$  which magnitude falls in the null level of confidence are discarded because they are considered noise (outliers). The remaining sets  $D_{\beta,\gamma}^\alpha$  are the representative intervals of feature  $\beta$  for class  $\alpha$ .

The percentage of values inside a representative interval of a feature  $\beta$  that are not overlapped with any other value inside all the representative intervals of the same feature  $\beta$  for all the remaining classes is the weight of the interval. The weight must be in the range  $[0, 1]$ . For example, if an interval of 30 values has 10 overlapped values, its weight is  $(30 - 10)/30$ . The different types of overlap between two intervals are shown in Fig. 1. In general, a given interval is overlapped by combinations of these base overlaps.



**Fig. 1.** The four types of overlap between two intervals: totally overlapped(A) where *non-overlapped area* = 0, partially left-overlapped (B) where *non-overlapped area* =  $(D_2 - I_2) - (D_1 - I_2)$ , partially right-overlapped (C) where *non-overlapped area* =  $(D_2 - I_2) - (D_2 - I_1)$  and partially center-overlapped (D) where *non-overlapped area* =  $(D_2 - I_2) - (D_1 - I_1)$

The obtained weights are used in the distance function of WIB-k:

$$Distance(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2 w(y_i)}$$

where  $x$  is the example to label and  $y$  is the labeled example stored in the concept description of WIB-K. If  $y_i$  falls within a representative interval, its weight will be the weight of the interval. If  $x_{i,j}$  does not fall in any representative interval, its weight will be the weight of the closest representative interval. The weights are normalized.

## 4 Results

### 4.1 Data Sets

We performed experiments and comparisons over several real world datasets from the UCI machine learning repository [14] in order to demonstrate the performance of the proposed algorithm. We selected databases with integer-valued features, without concerning about the type of the class. A brief description of the datasets is given in Table 3.

All the data sets have been randomly partitioned in ten disjoint sets for 10-fold cross validation. The same training and testing sets were used for all

the algorithms. Instances with missing values were removed. The compared algorithms were taken from Weka class library [8] and the parameters used are the default parameters, except for the  $K$  value that always was the same value used in WIB- $K$ .

**Table 3.** Description of the eight data sets used for experiments and comparisons

Name	Instances	Features	Classes
Balance Scale (BS)	625	4	3
Breast Cancer (BC)	699	11	2
CMC	1473	10	3
Dermatology (D)	366	34	6
Haberman (H)	306	4	2
Hayes Roth (HR)	162	6	3
Lung Cancer (LC)	32	57	3
TAE	151	6	3

### 4.2 Comparison Against Instance-Based and Weighted Instance-Based Algorithms

This subsection shows the results of the comparison of W-IB $K$  against other weighted and non weighted instance-based learning algorithms. IB1 and IB- $K$  are the implementations of the original instance-based learning algorithms proposed by D. Aha et al. [2]. dw-IB $K$ (1/d) and dw-IB $K$ (1-d) are the IB- $K$  algorithm weighted by the distance of the nearest neighbors. (1/d) represents that the weight is obtained from the inverse of the distance ( $1/distance$ ) whereas (1-d) means that the weight is obtained from the complement of the distance ( $1 - distance$ ). LWL is the implementation of the locally weighted learning algorithm proposed by Atkinson et al. [3]. Finally, K-Star is the implementation of the instance-based learner  $K^*$  proposed by J. C. Cleary and L. E. Trigg [5].

**Table 4.** Accuracy comparison between IB- $k$  and other weighted and non weighted instance-based learners

DB	K	WIB- $K$	IB1	IB- $K$	dwIB- $K$ (1/d)	dwIB- $K$ (1-d)	LWL	K-Star
BS	24	<b>90.396</b>	79.027	89.436	89.436	89.436	53.932	88.474
BC	5	<b>97.216</b>	96.04	97.079	<b>97.216</b>	<b>97.216</b>	92.09	95.458
CMC	16	<b>55.126</b>	43.312	48.404	48.264	48.4	48.47	49.553
D	1	90.238	<b>95.269</b>	<b>95.269</b>	<b>95.269</b>	<b>95.269</b>	82.642	94.126
H	27	<b>75.483</b>	65.709	74.494	73.537	74.494	72.505	71.204
HR	2	75.604	76.538	62.087	67.417	67.417	<b>79.395</b>	61.263
LC	1	<b>70</b>	48.333	48.333	48.333	48.333	56.666	56.666
TAE	1	<b>72.958</b>	63.583	62.291	62.291	62.291	52.916	64.291

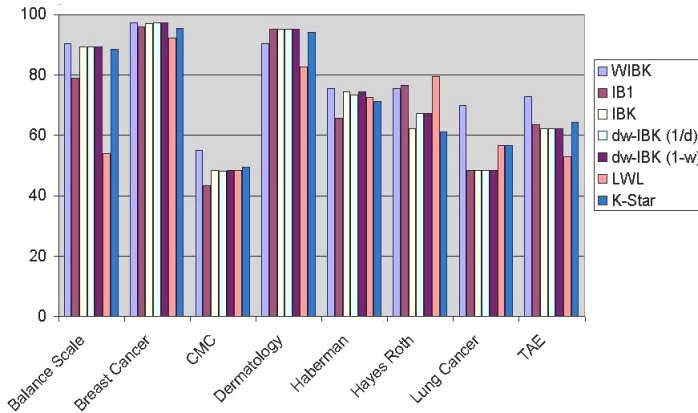


Fig. 2. Bar Graph of the results of Table 4

Table 4 shows the classification rate (in %) comparison between  $WIB-K$  and other weighted and non weighted instance-based learners.  $WIB-K$  has achieved higher accuracy for all data sets except Dermatology and Hayes Roth. Even for the Dermatology and Hayes Roth data sets,  $WIB-K$  is better than many classifiers. Thus, in 6 out of 8 data sets,  $WIB-K$  has achieved the best performance. Table 4 also shows that the proposed algorithm  $WIB-K$  is consistently better than the original algorithms  $IB1$  and  $IB-K$ .

Fig. 2 shows the classification rate with a bar graph. From the bar graph, it is clear that the proposed algorithm usually obtains superior results in terms of classification rate.

### 4.3 Comparison Against Well-Known Classifiers

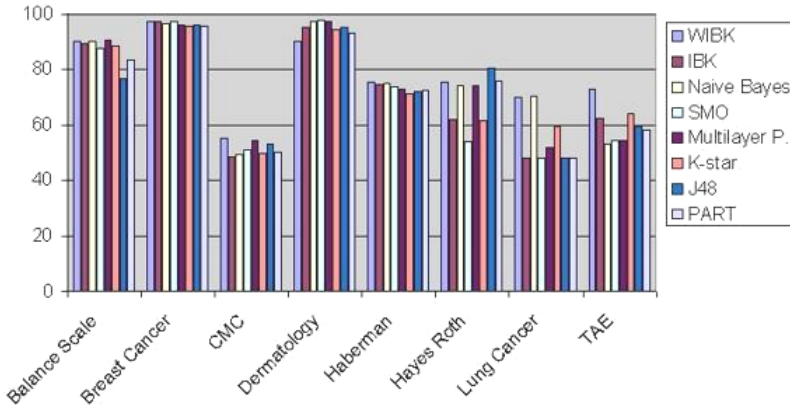
This subsection shows the results of the comparison of  $WIB-K$  against well-known and representative machine learning algorithms. NB is a naive bayes classifier that uses estimator classes [11]. SMO is the implementation of the algorithm to train a support vector classifier proposed by J. Platt [15]. MP is the implementation of a neural network that uses backpropagation to train. J48 is the implementation of the C4.5 decision tree proposed by R. Quinlan [16]. Finally, PART is a decision rule-based algorithm proposed by E. Frank and I. H. Witten [9].

Table 5 shows the accuracy (in %) achieved by  $WIB-K$  and other representative machine learning algorithms. The  $WIB-K$  algorithm has achieved the highest accuracy in the Breast Cancer, CMC, Haberman, and TAE data sets, and, for the remaining data sets,  $WIB-K$  performed better than many well-known machine learning algorithms. The other algorithms obtained the best result in at most one data set whereas  $WIB-K$  did it in four data sets. In 4 out of 8 data sets  $WIB-K$  achieved the best performance.

Fig. 3 shows a bar graph for the data presented in Table 5. In the bar graph we can see that the  $WIB-K$  algorithm is highly competitive.

**Table 5.** Accuracy comparison between IB-*k* and other well-known algorithms

DB Name	K	WIB-K	NB	SMO	MP	J48	PART
Balance Scale	24	90.39	90.04	87.68	<b>90.72</b>	76.64	83.52
Breast Cancer	5	<b>97.21</b>	96.33	97.07	96.04	96.04	95.46
CMC	16	<b>55.12</b>	49.28	50.98	54.51	53.22	50.10
Dermatology	1	90.23	97.48	<b>97.76</b>	97.48	95.25	93.29
Haberman	27	<b>75.48</b>	74.83	73.52	72.87	71.89	72.54
Hayes Roth	2	75.60	74.24	53.78	74.24	<b>80.30</b>	75.75
Lung Cancer	1	70	<b>70.37</b>	48.14	51.85	48.14	48.14
TAE	1	<b>72.95</b>	52.98	54.30	54.30	59.60	58.27



**Fig. 3.** Bar Graph for the results shown in Table 5

## 5 Conclusion and Future Work

In this paper we proposed a new weighted instance-based learning algorithm to perform classification of instances defined by integer valued features. This algorithm outputs one or more weights per feature for each class, and is noise tolerant. The weight is used in the distance function of the IB-K algorithm to improve accuracy rate.

The algorithm was tested on UCI databases of instances defined by integer attributes. The results indicate high competitiveness with respect to many well-known machine learning algorithms, as well as against weighted and non weighted instance-based learners.

The novelty of the algorithm relies in the approach to finds the intervals in which the value of a certain feature for a certain class falls, and obtains the weights directly from them. This approach opens new and interesting research paths. Knowing the interval in which the value of a certain feature for a certain class falls is important because it can give us more information about the data behavior.



Although this algorithm is restricted to integer-valued features, it is possible to apply it to real-valued features if, as a preprocessing step, the features are discretized [7]. Ordered nominal features can be directly converted into integers, however, the algorithm can not deal with non-ordered categorical features. Future work will focus on the search of a preprocessing scheme that allows WIB- $K$  to deal with all kind of features.

**Acknowledgements.** The first author acknowledges CONACYT the support provided through the grant for Master's studies number 201804. The first author also acknowledges Erika Danaé López Espinoza for her valuable comments.

## References

1. Aha, D.W.: Feature Weighting for Lazy Learning algorithms. In: Feature Extraction, Construction and Selection: A Data Mining Perspective, vol. 1, The American Statistical Association, Boston (1998)
2. Aha, D.W., Kibler, D., Albert, M.C.: Instance-Based Learning Algorithms. In: Machine Learning, vol. 6, pp. 37–66. Springer, Netherlands (1991)
3. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally Weighted Learning. In: Artificial Intelligence Review, vol. 11, pp. 11–73. Springer, Netherlands (1997)
4. Blansché, A., Gancarski, P., Korczak, J.J.: MACLAW: A modular approach for clustering with local attribute weighting. In: Pattern Recognition Letters, vol. 27, pp. 1299–1306. Elsevier, Amsterdam (2006)
5. Cleary, J.G., Trigg, L.E.:  $K^*$ : an instance-based learner using an entropic distance measure. In: Machine Learning: Proceedings of the Twelfth International Conference, pp. 108–114. Morgan Kaufmann, San Francisco (1995)
6. Cover, T.M., Hart, P.E.: Nearest Neighbor pattern classifier. In: IEEE Transactions on Information Theory, vol. 13, pp. 21–27. IEEE Transactions on Information Theory Society, Los Alamitos (1967)
7. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and Unsupervised Discretization of Continuous Features. In: Machine Learning: Proceedings of the Twelfth International Conference, Morgan Kaufmann, San Francisco (1995)
8. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
9. Witten, I.H., Frank, E.: Generating Accurate Rule Sets Without Global Optimization. In: Machine Learning: Proceedings of the Fifteenth International Conference, Morgan Kaufmann, San Francisco (1998)
10. Gartner, T., Flach, P.A.: WBCSVM: Weighted Bayesian Classification based on Support Vector Machines. In: Machine Learning: Proceedings of the Eighteenth International Conference, Morgan Kaufmann, San Francisco (2001)
11. George, H., Langley, P.: Estimating Continuous Distributions in Bayesian Classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, vol. 11, pp. 338–345. McGill University, Montreal (1995)
12. Kira, K., Rendell, L.: The feature selection problem: traditional methods and new algorithm. In: AAAI 1992. Proceedings of the Tenth National Conference on Artificial Intelligence, San Jose, California (1992)
13. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, Springer, Heidelberg (1994)

14. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, California (1998)
15. Plat, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: *Advances in Kernel Methods - Support Vector Learning*, MIT Press, Cambridge (1998)
16. Quinlan, J.R.: *Induction of Decision Trees*. In: *Machine Learning*, vol. 1, pp. 81–106. Springer, Netherlands (1986)
17. Smith, E.E., Medin, D.L.: *Categories and Concepts*. Harvard University Press, Cambridge (1981)
18. Tahir, T.A., Bouridane, A., Kurugollu, F.: Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier *Pattern Recognition Letters*, vol. 28, pp. 438–446. Elsevier, Amsterdam (2007)
19. De la Torre, A., Peinado, A.M., Rubio, J.A., Segura, J.C., Benítez, C.: Discriminative feature weighting for HMM-based continuous speech recognizers *Speech Communication*, vol. 38, pp. 267–286. Elsevier, Amsterdam (2002)
20. Weisstein, E.W.: The ANOVA test. Mathworld—A Wolfram web resource (2002), <http://mathworld.wolfram.com/ANOVA.html>
21. Wettschereck, D., Aha, D.W., Mohri, T.: A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms. In: *Artificial Intelligence Reviews*, vol. 5, pp. 273–314. Springer, Netherlands (1997)

# A Novel Information Theory Method for Filter Feature Selection

Boyan Bonev, Francisco Escolano, and Miguel Angel Cazorla

Department of Computer Science and Artificial Intelligence,  
Alicante University, P.O.B. 99, E-03080 Alicante, Spain  
{boyan, sco, miguel}@dccia.ua.es

**Abstract.** In this paper, we propose a novel filter for feature selection. Such filter relies on the estimation of the mutual information between features and classes. We bypass the estimation of the probability density function with the aid of the entropic-graphs approximation of Rényi entropy, and the subsequent approximation of the Shannon one. The complexity of such bypassing process does not depend on the number of dimensions but on the number of patterns/samples, and thus the curse of dimensionality is circumvented. We show that it is then possible to outperform a greedy algorithm based on the maximal relevance and minimal redundancy criterion. We successfully test our method both in the contexts of image classification and microarray data classification.

## 1 Introduction

Dimensionality reduction of the raw input variable space is a fundamental step in most pattern recognition tasks. Focusing on the most relevant information in a potentially overwhelming amount of data is useful for a better understanding of the data, for example in genomics [2] [21] [22]. A properly selected features set significantly improves classification performance. Thus, the removal of the noisy, irrelevant and redundant features is a challenging task.

There are two major approaches to dimensionality reduction: Feature Selection and Feature Transform. Whilst Feature Selection reduces the feature set by discarding the features which are not useful for some purpose (generally for classification), Feature Transform methods (also called feature extraction) build a new feature space from the original variables.

The literature differentiates among three kinds of Feature Selection: *Filter* methods [4], *Wrapper* methods [5], and *On-line* [6]. Filter Feature Selection does not take into account the properties of the classifier (it relies on statistical tests to the variables), while Wrapper Feature Selection tests different feature sets by building the classifier. Finally, On-line Feature Selection incrementally adds new features during the selection process.

Feature Selection is a combinatorial computational complexity problem. Algorithms must be oriented to find suboptimal solutions in a feasible number of iterations. Nevertheless, when there are thousands of features, Wrapper approaches become unfeasible. Among the Filter approaches, a fast way to evaluate

individual features is given by their relevance to the classification, by maximizing the mutual information between each variable and the classification output. As Guyon and Elisseeff state in [4], this is usually suboptimal for building a predictor, particularly if the variables are redundant. Conversely, a subset of useful variables may exclude many redundant, but relevant, variables. To overcome this limitation Peng et al. [7] minimize redundancy among the selected features set. Still a problem remains in the fact that these criteria are based on individual features, and this is due to the fact that estimating mutual information (and entropy) in a continuous multi-dimensional feature space is a hard task.

In this work, we overcome the latter problem by using Entropic Spanning Graphs to estimate Mutual Information [15]. The method's complexity does not depend on the number of dimensions, but on the number of samples. It allows us to estimate mutual information and thus maximize dependency between combinations of thousands of features and the class labels. We compare classification results to another Filter Feature Selection approach and perform an experiment on gene patterns with thousands of features.

This paper is structured as follows. In Section 2, the estimation of Mutual Information (Subsection 2.1) and Entropy (Subsection 2.2) are detailed. Then in Section 3, Feature Selection criteria and algorithms are explained and complexity is discussed. Finally experimental results are presentend in Section 4, and conclusions and future work are stated in Section 5.

## 2 Estimation of Mutual Information and Entropy

### 2.1 Mutual Information Estimation

Mutual Information (MI) is used in Filter Feature Selection as a measure of the dependency between a set of features  $S$  and the classification prototypes  $C$ . MI can be calculated in different ways. In [1], Neemuchwala et al. study the use of entropic graph for MI estimation. In our approach we calculate MI based on entropy estimation:

$$I(S; C) = \sum_{s \in S} \sum_{c \in C} p(s, c) \log \frac{p(s, c)}{p(s)p(c)} \quad (1)$$

$$= H(S) - H(S|C) \quad (2)$$

$$= H(S) + H(C) - H(S, C) \quad (3)$$

Using the Eq. 2 the conditional entropy  $H(S|C)$  has to be calculated. To do this,  $\sum (X|C = c)p(C = c)$  entropies have to be calculated, and this is feasible insofar  $C$  is discrete ( $C$  consists of the class labelling). On the other hand, using Eq. 3 implies estimating the joint entropy. In our experiments we used Eq. 2 because it is faster, due to the complexity of the entropy estimator, which depends on the number of samples as we will see in the following subsection.

### 2.2 Entropy Estimation

Entropy is a basic concept in information theory [8]. For a discrete variable  $Y$  with  $y_1, \dots, y_N$  (the set of values), we have:

$$\begin{aligned}
 H(Y) &= -E_y[\log(p(Y))] \\
 &= -\sum_{i=1}^N p(Y = y_i) \log p(Y = y_i).
 \end{aligned}
 \tag{4}$$

The estimation of the Shannon entropy of a probability density given a set of samples has been studied widely in the past [9][10][11][12][13][14]. Most current nonparametric entropy and divergence estimation techniques are based on estimation of the density function followed by the substitution of these estimates into the expression for entropy. This method has been widely applied to estimation of the Shannon entropy and it is called “plug-in” estimation [9]. Other methods of Shannon entropy estimation include sample spacing estimators, restricted to  $d = 1$ , and estimates based on nearest neighbor distances.

In [15] an alternative method for entropy and divergence estimation based on using entropic spanning graphs is presented. It is considered as a “non plug-in” method, because the entropy is directly estimated from a set of samples of the pdf, by-passing the non-parametric density estimation.

Among the “plug-in” methods, a widely used one is the Parzen’s Window. Each method has its own advantages and drawbacks: On the one hand in the Parzen’s Windows approach problems arise due to the infinite dimension of the spaces in which the unconstrained densities lie. Specifically: density estimator performance is poor without stringent smoothness conditions; no unbiased density estimators generally exist; density estimators have high variance and are sensitive to outliers; the high dimensional integration required to evaluate the entropy might be difficult. In contrast, the entropic graphs method does not estimate Shannon entropy directly and a new technique to obtain it must be developed. The main advantage of this approach is the possibility to work in a very high-dimensional space, in contrast to Parzen’s Windows, the complexity of which is quadratic with respect to the number of dimensions.

Entropic Spanning Graphs obtained from data to estimate Renyi’s  $\alpha$ -entropy [15] belong to the “non plug-in” methods of entropy estimation. Renyi’s  $\alpha$ -entropy of a probability density function  $f$  is defined as:

$$H_\alpha(p) = \frac{1}{1 - \alpha} \ln \int_z p^\alpha(z) dz
 \tag{5}$$

for  $\alpha \in (0, 1)$ . The  $\alpha$  entropy converges to the Shannon entropy  $-\int p(z) \ln p(z) dz$  as  $\alpha \rightarrow 1$ , so it is possible to obtain the second one from the first one.

A graph  $G$  consists of a set of vertices  $X_n = \{x_1, \dots, x_n\}$ , with  $x_n \in R^d$  and edges  $\{e\}$  that connect vertices in graph:  $e_{ij} = (x_i, x_j)$ . If we denote by  $M(X_n)$  the possible sets of edges in the class of acyclic graphs spanning  $X_n$  (spanning

trees), the total edge length functional of the Euclidean power weighted Minimal Spanning Tree is:

$$L_\gamma^{MST}(X_n) = \min_{M(X_n)} \sum_{e \in M(X_n)} |e|^\gamma \tag{6}$$

with  $\gamma \in (0, d)$  y  $|e|$  the euclidian distance between graph vertices.

The MST has been used as a way to test for randomness of a set of points. In [16] it was showed that in  $d$ -dimensional feature space, with  $d \geq 2$ :

$$H_\alpha(X_n) = \frac{d}{\gamma} \left[ \ln \frac{L_\gamma(X_n)}{n^\alpha} - \ln \beta_{L_\gamma, d} \right] \tag{7}$$

is an asymptotically unbiased, and almost surely consistent, estimator of the  $\alpha$ -entropy of  $p$  where  $\alpha = (d - \gamma)/d$  and  $\beta_{L_\gamma, d}$  is a constant bias correction depending on the graph minimization criterion, but independent of  $p$ . Closed form expressions are not available for  $\beta_{L_\gamma, d}$ ; only known approximations and bounds: (i) Monte Carlo simulation of uniform random samples on unit cube  $[0, 1]^d$ ; (ii) Large  $d$  approximation:  $(\gamma/2) \ln(d/(2\pi e))$  in [17].

We can estimate  $H_\alpha(p)$  for different values of  $\alpha = (d - \gamma)/d$  by changing the edge weight exponent  $\gamma$ . As  $\gamma$  modifies the edge weights monotonically, the graph is the same for different values of  $\gamma$ , and only the total length in expression [7] needs to be recomputed.

Entropic spanning graphs are suitable for estimating  $\alpha$ -entropy with  $\alpha \in [0, 1[$ , so Shannon entropy can not be directly estimated with this method. In [18] relations between Shannon entropy and Rényi entropies of integer order are discussed. For any discrete probability  $n$ -points distribution for which the Rényi entropies of order two and three are known he provides a lower and an upper bound for the Shannon entropy. In [19] Mokkaem constructed a nonparametric estimate of the Shannon entropy from a convergent sequence of  $\alpha$ -entropy estimates.

The value of  $H_\alpha$  for  $\alpha = 1$  is approximated by means of a continuous function that captures the tendency of  $H_\alpha$  in the environment of 1. Such a function is a monotonous decreasing one, and by means of a dichotomic search we find the  $\alpha^*$  value that is used for extrapolating the correct entropy value. In [20] the process is explained in more detail, and it is experimentally verified that  $\alpha^*$  is constant for a fixed number of samples and dimensions, and for different covariance matrixes.

### 3 Feature Selection Criteria and Algorithms

There are different Filter Feature Selection criteria for selecting or discarding a feature or a feature set. In [7], Peng et al. study the possibility maximize the dependency between the feature set  $S$  and the prototypes  $C$ :  $\max I(S; C)$ , called Max-Dependency criterion. Eq. [8] formulates the maximization objective for selecting the  $m$ -th feature from the  $X - S_{m-1}$  set of features which still are not selected.

$$\max_{x_j \in X - S_{m-1}} I(S_{m-1} \cup \{x_j\}; c) \tag{8}$$

In [7] this criterion is found to be unfeasible because the entropy estimation in high-dimensional feature spaces is very hard, and yields poor results, due to the way they estimate entropy. So instead of estimating mutual information in a multidimensional space, they maximize the relevance  $I(x_j; c)$  of each individual feature  $x_j$  and at the same time minimize the redundancy between  $x_j$  and the rest of selected features  $x_i \in S, i \neq j$ . This is the Max-Relevance Min-Redundance (mRMR) criterion, Eq. [9]

$$\max_{x_j \in X - S_{m-1}} \left[ I(x_j; c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_j; x_i) \right] \tag{9}$$

In this work, we state that the Max-Dependency criterion is feasible, even in very high-dimensional feature spaces. The complexity of MI estimation, explained in Section [2], depends on the Entropic Spanning Graph construction, which has a  $O(s \log(s))$  order, where  $s$  is the number of samples. Also, the accuracy of the estimations does not depend on the number of dimensions.

Another issue, is the way features combinations are generated. An exhaustive search among the features set combinations would present a  $O(n!)$  combinatorial complexity, where  $n$  is the total number of features. For our experiments we have used a Greedy Forward Feature Selection algorithm, which starts from a small feature set, and adds one feature at a time.

Therefore, with the mRMR criterion each iteration would consist of calculating the MI between a feature and the prototypes, as well as the MI between that feature and each one of the already selected ones (see Eq. [9]). Such a search performs  $\sum_{i=1}^n i(n-i+1)$  estimations of the MI, which has a  $O(n^3+n^2+n)$  computational complexity. Using the Max-Dependency criterion instead, requires just one MI calculus per iteration. The total number of MI estimations is  $\sum_{i=1}^n n-i+1$ , which has a  $O(n^2+n)$  computational complexity.

## 4 Experiments

### 4.1 Image Data

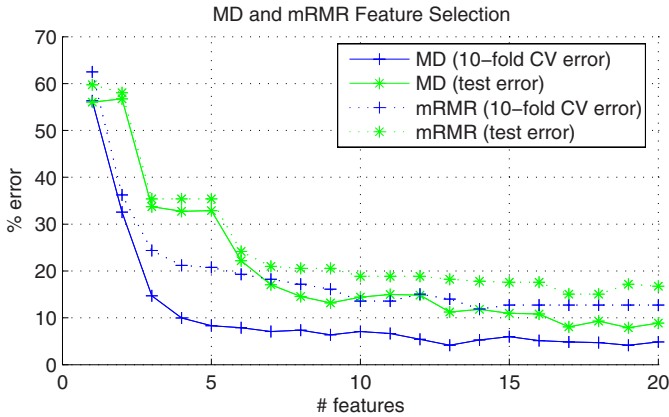
Two experiments on real data are presented in this paper. The first of them compares the Max-Dependency and the mRMR criteria. The data set consists of a set of 721 images (samples), labeled with 6 different classes. Each sample has 48 features, which come from some basic filters responses, like color filters, corner and edge detectors, and range information. Such a configuration is useful for image classification and image registration purposes. On Fig. [1] we have represented image registration results for a few outdoor images.

In Fig. [2] we show the classification errors of the feature sets selected with both criteria. A Nearest Neighbour classification was evaluated because the number of samples is not very high. Only 20 selected features are represented, as for larger features set the error does not decrease. The 10-fold Cross Validation error is represented, as well as a test-error, which was calculated using an additional test-set of images, containing 470 samples.



**Fig. 1.** Image registration experiment on different test images. The first row contains test images. Each one of them is associated to some image of the training set, shown on the second row. The training set contains 721 images taken during an indoor-outdoor walk. The test set has not been used for the feature selection process and it is taken during a different walk following a similar trajectory. The amount of low-level filters selected for building the classifier is 13, out of 48 in total.

With mRMR, the lowest CV error (8.05%) is achieved with a set of 17 features, while with Max-Dependency a CV error of 4.16% is achieved with a set of 13 features. The test-errors have similar tendencies.



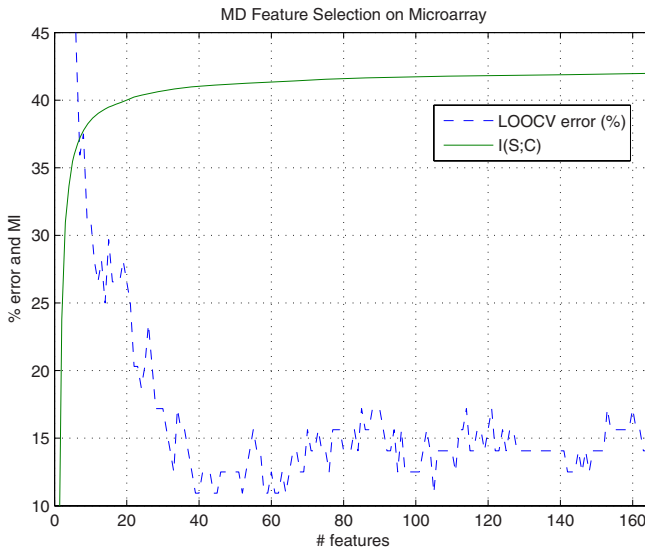
**Fig. 2.** Feature Selection performance on image histograms data with 48 features. Comparison between the Maximum-Dependency (MD) and the Minimum-Redundancy Maximum-Relevance (mRMR) criterions.

In Peng et al. [7], the experiments yielded better results with the mRMR criterion than with the Max-Dependency criterion. Contrarily, we obtain better performance using Max-Dependency. This may be explained by the fact that the entropy estimator we use does not degrade its accuracy as the number of dimensions increases.



### 4.2 Microarray Data

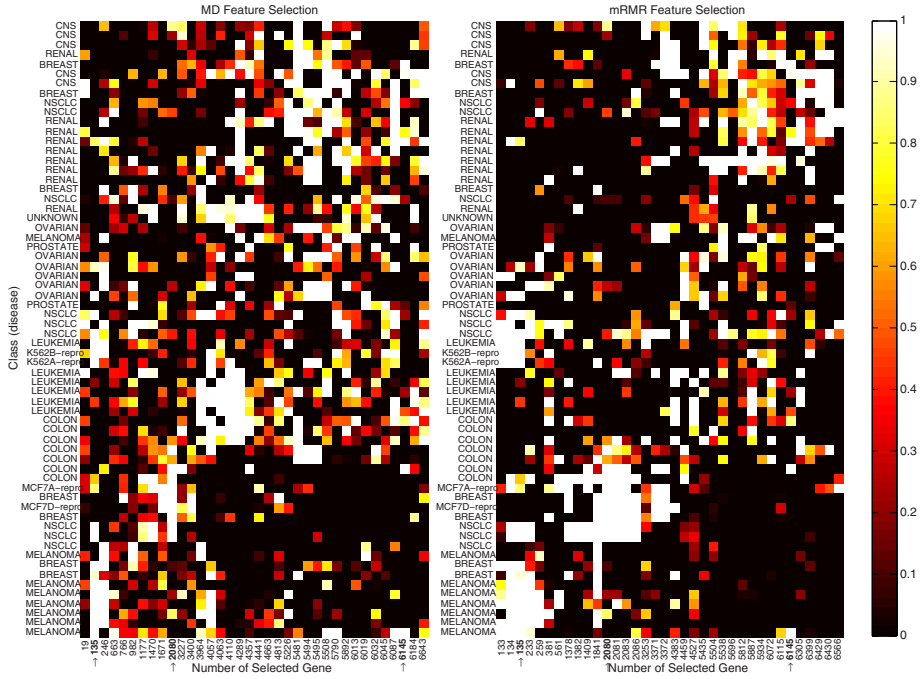
In order to illustrate the dimensionality independence of the the entropy estimator we use, we performed another experiment on the well-known NCI60 DNA microarray dataset. It contains only 60 samples, labeled with 14 different classes of human tumor diseases. Each sample has 6380 dimensions (genes). The purpose of Feature Selection is to select those genes which are useful for disease classification/prediction. An example can be seen on Fig 4 where we have represented the first 36 features selected by both criteria discussed in this paper. The rows represent the samples, or diseases. The columns represent features. The intensity of the colour is the level of expression of a gene for a given disease. In this figure we can see that MD and mRMR select different genes.



**Fig. 3.** Feature Selection performance on microarray data with 6380 features. The FFS algorithm using the Maximum Dependency criterion obtained the lowest LOOCV error with a set of 39 features. The function that is maximized (the mutual information) is also represented.

In Fig. 3, we show the Leave One Out Cross Validation errors<sup>1</sup> for the selected feature subsets, using the Max-Dependency criterion. Only the best 220 genes (out of 6380) are on the X axis, and it took about 24 hours on a PC with Matlab to select them. During this time MI was calculated  $\sum_{i=1}^{220} (6380 - i + 1) = 1,385,670$  times.

<sup>1</sup> LOOCV measure is used when the number of samples is so small that a test set cannot be built. It consists of building all possible classifiers, each time leaving out only one sample for test.



**Fig. 4.** Feature Selection on the NCI DNA microarray data. The MD (on the left) and mRMR (on the right) criteria were used. Features (genes) selected by both criteria are marked with an arrow.

In [3] an evolutionary algorithm is used for feature selection and the best LOOCV error achieved is 23,77% with a set of 30 selected features. In our experiment we achieve a 10,94% error with 39 selected features.

## 5 Conclusions and Future Work

In this paper we presented a Filter Feature Selection approach based on Mutual Information. The Mutual Information estimation does not depend on the number of features, but it depends n-logarithmically on the number of samples. Therefore this approach is useful for high-dimensional patterns, such as DNA microarray data. In contrast to Wrapper approaches, this Filter approach does not rely on minimizing the classification error, but on maximizing MI of sets of features and class labels. However as a consequence of this, the classification error actually decreases.

Finally, we obtain better results by evaluating MI (Max-Dependency) for the entire feature subsets, than the criterion of Min-Redundancy Max-Relevance.

In the future we want to explore Feature Selection algorithms different than FFS. This algorithm starts selecting small feature subsets, but with our approach it would not be hard to start from larger feature subsets and remove the less informative ones.

**Acknowledgments.** This research is funded by the project DPI2005-01280 from the Spanish Government.

## References

1. Neemuchwala, H., Hero, A., Carson, P.: Image registration methods in high dimensional space. *International Journal on Imaging* (2006)
2. Sima, C., Dougherty, E.R.: What should be expected from feature selection in small-sample settings. *Bioinformatics* 22(19), 2430–2436 (2006)
3. Jirapech-Umpai, T., Aitken, S.: Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics* 6, 148 (2005)
4. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
5. Blum, A.L., Langley, P.: Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence* (1997)
6. Perkins, S., Theiler, J.: Online Feature Selection using Grafting. In: *ICML 2003. Proceedings of the Twentieth International Conference on Machine Learning*, Washington DC (2003)
7. Peng, H., Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(8) (2005)
8. Cover, T., Thomas, J.: *Elements of Information Theory*. J. Wiley and Sons, Chichester (1991)
9. Beirlant, E., Dudewicz, E., Gyorfi, L., Van der Meulen, E.: Nonparametric Entropy Estimation. *International Journal on Mathematical and Statistical Sciences* 5(1), 17–39 (1996)
10. Paninski, I.: Estimation of Entropy and Mutual Information. *Neural Computation* 15(1) (2003)
11. Viola, P., Wells-III, W.M.: Alignment by Maximization of Mutual Information. In: *5th Intern. Conf. on Computer Vision*, IEEE, Los Alamitos (1995)
12. Viola, P., Schraudolph, N.N., Sejnowski, T.J.: Empirical Entropy Manipulation for Real-World Problems. *Adv. in Neural Infor. Proces. Systems* 8(1) (1996)
13. Hyvarinen, A., Oja, E.: Independent Component Analysis: Algorithms and Applications. *Neural Networks* 13(4-5), 411–430 (2000)
14. Wolpert, D., Wolf, D.: Estimating Function of Probability Distribution from a Finite Set of Samples, Los Alamos National Laboratory Report LA-UR-92-4369, Santa Fe Institute Report TR-93-07-046 (1995)
15. Hero, A.O., Michel, O.: Applications of spanning entropic graphs. *IEEE Signal Processing Magazine* 19(5), 85–95 (2002)
16. Hero, A.O., Michel, O.: Asymptotic theory of greedy approximations to minimal  $k$ -point random graphs. *IEEE Trans. on Infor. Theory* 45(6), 1921–1939 (1999)
17. Bertsimas, D.J., Van Ryzin, G.: An asymptotic determination of the minimum spanning tree and minimum matching constants in geometrical probability. *Operations Research Letters* 9(1), 223–231 (1990)
18. Zyczkowski, K.: Renyi Extrapolation of Shannon Entropy. *Open Systems and Information Dynamics* 10(3), 298–310 (2003)

19. Makkadem, A.: Estimation of the entropy and information of absolutely continuous random variables. *IEEE Trans. on Inform. Theory* 35(1), 193–196 (1989)
20. Peñalver, A., Escolano, F., Sáez, J.M.: EBEM: An Entropy-based EM Algorithm for Gaussian Mixture Models. *ICPR*, 451–455 (2006)
21. Xing, E.P., Jordan, M.I., Karp, R.M.: Feature selection for high-dimensional genomic microarray data. In: *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 601–608 (2001)
22. Gentile, C.: Fast Feature Selection from Microarray Expression Data via Multiplicative Large Margin Algorithms. In: *Proceedings NIPS* (2003)

# Building Fine Bayesian Networks Aided by PSO-Based Feature Selection

María del Carmen Chávez, Gladys Casas, Rafael Falcón,  
Jorge E. Moreira, and Ricardo Grau

Computer Science Department, Central University of Las Villas  
Carretera Camajuaní km 5 ½ Santa Clara, Cuba  
{mchavez, gladita, jmoreira, rfalcon, rgrau}@uclv.edu.cu

**Abstract.** A successful interpretation of data goes through discovering crucial relationships between variables. Such a task can be accomplished by a Bayesian network. The dark side is that, when lots of variables are involved, the learning of the network slows down and may lead to wrong results. In this study, we demonstrate the feasibility of applying an existing Particle Swarm Optimization (PSO)-based approach to feature selection for filtering the irrelevant attributes of the dataset, resulting in a fine Bayesian network built with the K2 algorithm. Empirical tests carried out with real data coming from the bioinformatics domain bear out that the PSO fitness function is in a straight concordance to the most widely known validation measures for classification.

**Keywords:** Bayesian network, feature selection, classification, particle swarm optimization, validation measures.

## 1 Introduction

Bayesian networks are a powerful knowledge representation and reasoning tool that behave well under uncertainty [2], [3], [4]. A Bayesian network is a directed acyclic graph (DAG) with a probability table associated to each node. The nodes in a Bayesian network represent propositional variables in a domain and the edges between them stand for the dependency relationships among the variables. When constructing Bayesian networks from databases, the most common approach is to model a set of attributes as network nodes [2], [3].

Learning a Bayesian network from data within the Bioinformatics field is often expensive, since we start considering quite a few numbers of variables, which has a detrimental direct impact on the building time it is required to finish shaping the network. Recall you have to make up both the nodes as well as their associated probabilities tables. This triggers the need to look for attribute reduction algorithms.

It is the aim of this study to demonstrate that an existing Particle Swarm Optimization (PSO)-based algorithm to feature selection can be used as a starting point to compute the most suitable reduct (i.e., the minimal number of attributes which correctly characterizes the dataset) from which the network can be built upon,

therefore dropping the irrelevant attributes present in the dataset. This approach exploits the high convergence rate inherent to the PSO algorithm and yields a direct measure (fitness function value) of the reduct's quality.

From this point on, you can build a Bayesian network with sound good characteristics by using any of the traditional algorithms previously mentioned in the literature, such as the K2 method, although in classification task it is possible to use the NB (Naïve Bayes), or TAN (Tree Augmented Bayes Network), etc. [14]. Such algorithms are responsible of uncovering the hierarchical relationships between the attributes already included in our reduct. Our claim is empirically supported by a statistical concordance between the aforementioned fitness value and several widely known measures that serve the purpose of evaluating the performance of the classification systems, in this case the Bayesian network model.

The study is structured as follows: Section 2 dwells on the main details of a Bayesian network whereas Section 3 summarizes the key ideas behind the PSO meta-heuristic and the PSO-based feature selection algorithm. The experiments carried out to test the feasibility of our contribution are properly included in Section 4. Finally, some conclusions and comments are provided.

## 2 An Overview to Bayesian Networks

A Bayesian network (also called “probabilistic network”) is formally a pair  $(D, P)$  where  $D$  is a Directed Acyclic Graph (DAG),  $P = \{p(x_1|\pi_1), \dots, p(x_n|\pi_n)\}$  is a set of  $n$  Conditional Probabilistic Distributions (CPD), one for each variable  $x_i$  (nodes of the graph), and  $\pi_i$  is the set of parents of node  $x_i$  in  $D$ . The set  $P$  defines the associated joint probabilistic distribution as shown in Equation 1.

$$p(x) = \prod_{i=1}^n p(x_i|\pi_i) \quad x = (x_1, x_2, \dots, x_n) \quad (1)$$

Before the network can be used, it is necessary to undertake a learning stage which, in turn, can be divided into two steps: (1) defining the structure (topology) of the network and (2) setting the values of the parameters involved in the probability distribution associated with each node.

Making up the network's topology is a tough process, since you would have to consider all possible combinations of attributes in order to determine the hierarchical relationships between them (i.e., which of them are parents of which nodes). The usual behavior encountered in literature is to use heuristic search procedures that browse all throughout the search space attempting to find a fairly good reduct. We do not step aside of this state of mind but come up with a novel approach using PSO which offers a straight measure (fitness function value) that is in correlation to the classical measures for the assessment of the performance of classifiers. Our claim is that the set of attributes bore by the PSO-based feature selection algorithm is an excellent starting point for building the network topology.

### 3 Particle Swarm Optimization

PSO is an evolutionary computation technique proposed by Kennedy and Eberhart [9] [10] [11] [12]. This concept was motivated by the simulation of social behavior of biological organisms such as bird flocks or fish schools. The original intent was to reproduce the graceful but unpredictable movement of bird flocking. The PSO algorithm takes very seriously the idea of sharing information between the individuals (particles) so as to solve continuous optimization problems, although specific versions for discrete optimization problems have been developed [8] [13] [15] [16] [24].

#### 3.1 PSO Fundamentals and Notations

PSO is initialized with a population of random solutions, called ‘particles’. Each particle is treated as a point in an  $S$ -dimensional space. The  $i$ -th particle is represented as  $X_i = (x_{i1}, x_{i2}, \dots, x_{iS})$  and keeps track of its coordinates in the problem space. PSO also records the best solution of all the particles (*gbest*) achieved so far, as well as the best solution (*pbest*) reached by each particle thus far. The best previous position of any particle is recorded and represented as  $P_i = (p_{i1}, p_{i2}, \dots, p_{iS})$ . The velocity of the  $i$ -th particle is denoted as a vector  $V_i = (v_{i1}, v_{i2}, \dots, v_{iS})$ . The individual’s velocity is updated following the criteria below:

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id}) \quad (2)$$

where  $v_i$  is the current velocity of the  $i$ -th particle,  $p_i$  is the position reaching the best fitness value visited by the  $i$ -th particle and  $g$  is the particle having the best fitness among all the particles,  $d = 1, 2, \dots, S$ . Additionally,  $w$  is the inertia weight which may be fixed before the algorithm execution or may dynamically vary as the algorithm executes. The acceleration constants  $c_1$  and  $c_2$  in (2) represent the weighting of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions, respectively. Some versions of the PSO meta-heuristic confine the particle’s velocity on each dimension to a specified range limited by  $V_{\max}$ . The performance of each particle is computed according to a predefined fitness function. Finally, each particle is updated as in (3):

$$x_{id} = x_{id} + v_{id} \quad (3)$$

#### 3.2 Moving on to the Formal Algorithm

The formal algorithm drawn from [24] is the following:

```

Given:
m: the number of particles;
c1, c2: positive acceleration constants;
w: inertia weight
MaxV: maximum velocity of particles
MaxGen: maximum generation
MaxFit: maximum fitness value
Output:

```

```

Pgbest: Global best position
Begin
Swarms {xi, vi} = Generate (m); /* Initialize a
population of particles with random positions and
velocities on S dimensions */
Pbest(i) = 0; i = 1, . . . ,m, d = 1, ..., S
Gbest = 0; Iter = 0;
While(Iter < MaxGen and Gbest < MaxFit)
{For(every particle i)
  {Fitness(i) = Evaluate(i);
   IF(Fitness(i) > Pbest(i))
    {Pbest(i) = Fitness(i); pi= xi; }
   IF(Fitness(i) > Gbest)
    {Gbest = Fitness(i); gbest = i;}
  }
For(every particle i)
  Update its velocity using (2)
  IF(vid > MaxV) {vid = MaxV;}
  IF(vid < -MaxV) {vid = -MaxV;}
  Update its position using (3)
}
}
  Iter = Iter + 1;
} /* r1 and r2 are two random numbers in [0, 1] */
Return P_{gbest}
End

```

### 3.3 PSO and Rough Set Theory to Feature Selection

In this section we confine ourselves to briefly describe the algorithm depicted in [24]. The optimal feature selection problem can be approached via PSO in the following way.

Assuming that we have  $N$  total features and that each feature may or may not be a part of the optimal redoubt it adds up a total of  $2^N$  possible vectors. Each vector is represented by a particle in the described approach. Over time, they change their position, communicate with each other and search around the local best and global best positions.

The particle's position was represented as a binary bit string of length  $N$  (the total number of attributes). Has a bit a value of 1, it indicates that the current attribute is selected to make up the redoubt; otherwise the bit is reset.

The velocity of each particle varies from 1 to  $V_{\max}$  and can be semantically interpreted as the number of features that must be changed in order to match that of the global position. A thorough explanation can be found at [24].

Once the particle's velocity has been updated by (2), the particle's position is also updated by comparing the current velocity with the number of different bits between the current particle and *gbest*. A semantic meaning leads to the update of the particle's position.

This algorithm also limits the maximum speed  $V_{\max}$  initially to the range  $[1, N]$  but it was afterwards shortened to the interval  $[1, N/3]$ . By limiting the maximum speed, the particle can not fly too far away from the optimal solution.



One of the most important components of the PSO technique is the fitness function which is used to evaluate how good the position of each particle is. In this case, the selected fitness function is outlined in (4).

$$Fitness = \alpha * \gamma_R(D) + \beta * \frac{|C| - |R|}{|C|} \quad (4)$$

where  $\gamma_R(D)$  is a rough set based measure known as “quality of classification” applied to the conditional set of attributes  $R$  with respect to  $D$  [19], [24];  $|R|$  is the number of bits set to one in the particle i.e. the length of the redoubt represented by the particle whereas  $|C|$  is the total number of features. It is also worth stressing that  $\alpha$  and  $\beta$  are two parameters corresponding to the importance of the quality of classification and the redoubt length, respectively with the constraints that  $\alpha \in [0, 1]$  and  $\beta = 1 - \alpha$ .

Another remarkable improvement made to the original PSO algorithm is the dynamic variation of the inertia weight by lowering it as the number of iterations increase. This guarantees that, at the beginning, a higher value of  $w$  brings about a faster exploration along the search space. As the algorithm is executed, this initial value is decreased to a predefined threshold, encouraging the local exploration.

## 4 Empirical Results

In this section we are about to introduce the dataset coming from the bioinformatics domain we have used to support our viewpoint (that by previously filtering a dataset in terms of the relevant attributes and using a sound indicator of the goodness of the reduct, comparable results with other traditional validation measures for classification are achieved). Subsequently, we will delve on the experiments carried out as well as the statistical tests to check their validity.

### 4.1 An Overview of the Reverse Transcriptase Protein Dataset

Proteins are the primary components of living things. If they are the workhorses of the biochemical world, nucleic acids are their drivers because they control the proteins’ action. All of the genetic information in any living creature is stored in deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) components, which are polymers of four simple nucleic acid units, called nucleotides.

Four nucleotides can be found in DNA, each of them consisting of three parts: one of two-base molecules (a purine or a pyrimidine) plus a sugar (ribose in RNA and deoxyribose DNA) and one or more phosphate groups. The purine nucleotides are adenine (A) and guanine (G) while the pyrimidines are cytosine (C) and thymine (T). Nucleotides are sometimes called bases and, since DNA is comprised by two complementary strands bonded together, these units are often called base-pairs. The length of a DNA sequence is often measured in thousands of bases, abbreviated kb. Nucleotides are generally abbreviated by their first letter and appended into sequences e.g., CCTATAG [6].

The task under consideration here is the prediction of mutations of the DNA sequence of Reverse Transcriptase (RT) protein which was obtained at the Stanford University. Such dataset involves 603 base-pairs in 419 sequences (cases). It is publicly available online at <http://hivdb.stanford.edu/cgibin/PIResiNote.cgi>

In order to set the values for each attribute, the list of mutations described in the dataset was used along with the reference strain HXB2. In particular, for each sample, the original sequence was reconstructed by replacing each mutation with the reverse transcriptase genes of HXB2 (Gene Bank access number K03455). In this way, each sequence that was built has no gaps or areas with loss of information and therefore, the “unknown” attribute value is no longer required [7] [17] [18].

This modified dataset is to be recoded with the results of the Research Group on Genomics Algebra [20] [21] [22], i.e. G ↔ 00; A ↔ 01; T ↔ 10 and C ↔ 11 for a grand total of 1206 attributes. The number of decision classes (values of the dependent variable) was set to three before applying the K-means clustering algorithm so as to group the DNA sequences into three well-defined clusters. This number was set empirically [5].

Table 1 pictures the search process carried out with the PSO-based feature selection approach for the reverse transcriptase protein of the HIV virus. In this case, up to 10 000 runs of the algorithm were performed so as to observe the behavior of the algorithm.

A shallow look at the results shown at Table 1 allows remarking how the fitness value is stabilized as the number of iterations increase and the obtained reduct remains the same. It draws us to think of the possibility of using such reduct as a starting point for the construction of the Bayesian network. Notice that, after a few number of iterations, the PSO approach is able to find a reduct with a high fitness value (Iterations = 30, Fitness = 0.978). When the number of iterations is 500, the algorithm behavior is stabilized. The fitness is 0.997 and the size of the optimal reduct is 12.

**Table 1.** Execution of the PSO- based feature selection algorithm. When the number of iterations is 500, the algorithm behavior is stabilized.

Iterations	Best Solution	Fitness Value	Redoubt Length
30	5 22 25 29 45 48 50 53 54 61 68 69 78 86 93 96 100 112 115 121 123 124 134 145 146 147 159 164 165 172 174 176 179 180 183 190 194 198 212 213 219 222 227 229 246 253 254 257 260 262 264 266 267 268 272 279 284 296 297 304 305 311 313 316 319 321 324 333 336 337 341 343 346 348 353 359 360 363 366 367 374 375 377 390 391 395 398 401 404	0.978	85
50	5 22 25 29 48 53 54 61 68 69 78 86 93 100 121 123 124 134 145 146 164 165 172 174 176 179 180 183 194 198 212 219 222 227 229 246 254 257 260 262 264 267 268 279 284 296 297 304 305 311 313 316 319 321 324 333 337 341 343 348 353 359 360 363 366 367 374 377 390 391 395 398 401 404	0.981	74
100	5 22 29 48 53 61 68 86 93 100 124 134 145 146 164 172 212 254 260 262 264 268 279 284 305 311 313 316 319 324 333 341 343 348 353 359 360 363 366 374 390	0.989	41
500	5 61 93 134 260 262 324 341 343 353 360 366	0.997	12
1000	5 61 93 134 260 262 324 341 343 353 360 366	0.997	12
10000	5 61 93 134 260 262 324 341 343 353 360 366	0.997	12

Our experiment was configured as follows: the PSO-based algorithm was run fifteen times and both the value of the fitness function and the yielded reduct were stored for each run with 500 iterations each. The size of the swarm was set to 20 particles. The balancing parameters were chosen as  $\alpha = 0.9$  and  $\beta = 0.1$ .

The second stage of the experiment was the building of the Bayesian networks, each of them using a single reduct from the ones obtained in the previous step. The K2 machine learning algorithm for building Bayesian networks was utilized in the framework of the WEKA environment [1] [2] [3] [25]. The first parameter in the BN is the maximum number of parents. We set up the network so as to use two parents per node, but it is possible to set this parameter to one (1), which automatically turns it into a Naive Bayes classifier. When set to 2, a TAN is learned and when the number of parents is greater than two, then a Bayes Augmented Network (BAN) is learned. Another parameter is the order of the variables. We assume that the variables are randomly ordered. The third parameter is the score type, which determines the measure used to assess the quality of the network structure. You may pick up one of the following list: Bayes, BDeu, Minimum Description Length (MDL), Akaike Information Criterion (AIC) and Entropy. For our experiments, we selected the Bayes measure [3] because it is reliable although any of the remaining ones could also be employed.

After running the Bayesian network against the previously outlined bioinformatics dataset, a group of measures for validating the classification is reported. Among them: the accuracy of the classification, the area under the ROC curves (AUC) and precisions per class. It is known that the ROC curve contains all the information about the False Positive and True Positive rates [25]. The outcome of the experiment is portrayed in Table 2.

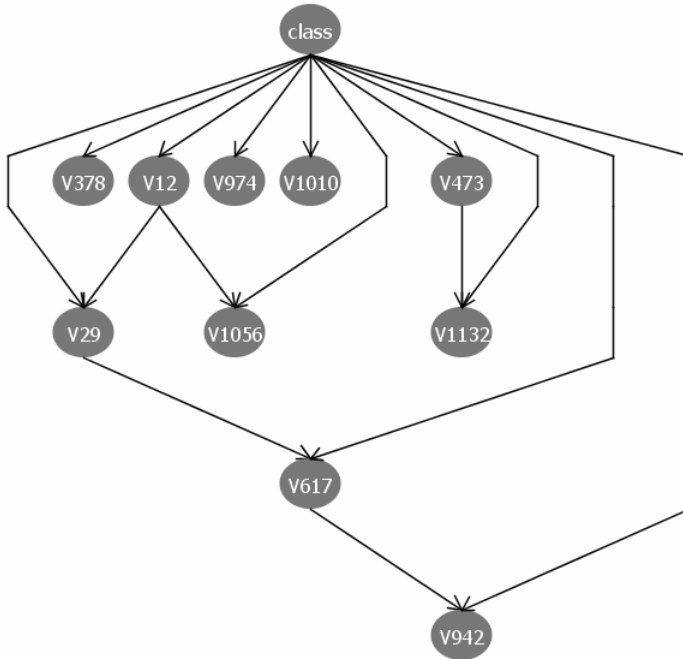
Our next step is to find out whether there exists a concordance between the fitness value and the rest of the measures (columns) in Table 2. For that purpose, we set up a Kendall test [23].

**Table 2.** Fitness and overall accuracy as well as AUC and precision per class

Iteration	Fitness	Accuracy	AUC <sub>1</sub>	AUC <sub>2</sub>	AUC <sub>3</sub>	Prec <sub>1</sub>	Prec <sub>2</sub>	Prec <sub>3</sub>
1	0.9873	98.09	0.999	1	1	0.902	0.977	1
2	0.9958	94.27	0.992	0.986	0.995	0.915	0.894	0.975
3	0.9080	94.51	0.992	0.985	0.994	0.913	0.883	0.987
4	0.9955	95.22	0.997	0.994	0.998	0.936	0.916	0.975
5	0.9970	98.32	0.998	0.999	1	0.957	0.969	0.996
6	0.9975	98.56	0.999	1	1	0.958	0.977	0.996
7	0.9972	94.27	0.969	0.987	0.999	0.81	0.89	0.996
8	0.9965	97.85	0.996	0.999	1	0.977	0.948	0.996
9	0.9958	96.65	1	0.997	0.998	1	0.925	0.983
10	0.9965	98.56	0.996	0.999	0.996	0.978	0.969	0.996
11	0.9958	96.89	0.996	0.958	1	0.932	0.947	0.988
12	0.9965	97.61	0.997	0.998	1	0.955	0.947	0.996
13	0.9960	97.61	0.995	0.999	1	0.976	0.941	0.996
14	0.9972	98.56	0.998	1	1	0.939	1	0.988
15	0.9960	97.85	0.997	0.999	1	0.956	0.962	0.992

**Table 3.** Mean Ranks according to Kendall test. Kendall coefficient (W).

Iterations	Fitness-Accuracy	Fitness-Accuracy-AUC	Fitness-Accuracy-AUC-Precision
1	9.50	5.70	5.75
2	12.75	13.20	13.19
3	14.00	14.10	13.81
4	12.50	10.90	11.38
5	4.00	4.70	4.88
6	1.50	2.50	3.13
7	8.50	10.80	11.00
8	6.25	6.70	6.06
9	11.00	8.90	8.69
10	4.00	7.40	6.06
11	10.50	10.20	10.13
12	7.25	7.10	7.13
13	8.50	8.00	7.38
14	2.25	3.20	4.56
15	7.50	6.60	6.88
Kendall's W	0.754	0.611	0.552



**Fig. 1.** The structure of the Bayesian network for the reduct computed in the iteration number 6 which consists of 10 nodes. The learned Bayesian network represents the information well, allowing the inference of mutations in reverse transcriptase sequences.

Kendall's test was run with several measure groups: Fitness-Accuracy, Fitness-Accuracy-AUC and Fitness-Accuracy-AUC-Precision. The results appear in Table 3. For each run, a list with the mean ranks of iterations and Kendall's W coefficients are shown. There exists a satisfactory concordance between the fitness of PSO method and the Accuracy ( $W = 0.754$ ). The W value for Fitness-Accuracy-AUC is 0.611 and if we include Precision for class as a fourth estimator, we obtain  $W = 0.555$ , in all cases greater than 0.5. So we conclude that there exists an adequate concordance between Fitness and the mentioned classification gauges.

The best results are always obtained in the iteration number 6 (the lowest rank). In Table 2 it can be seen that in this case we effectively yield the best fitness and network's performance. Remark that in such iteration we obtain a reduct with size 10, more simplified than the one initially produced. The corresponding Bayesian Network is portrayed in Fig. 1.

The nodes of the network can be interpreted through the operation module 6 (remember that each codon is represented as 6 binary digits). For instance, position  $378 = 63 * 6$  corresponds to position 6 of codon 63; position  $974 = 162 * 6 + 2$  corresponds to position 2 of codon 163; position 473 corresponds to position 5 of codon 79. The arcs stand for the dependences. So, the final class depends on all positions. Position 12 interacts with position 29 and 1056 whereas position 473 interacts with position 1132 and position 617 interacts with position 942.

After completing the network's topology, each node is assigned a suitable conditional probabilities table. The WEKA environment was utilized for this purpose. The resultant accuracy was 98.5% as stated above.

## 5 Conclusions

In this paper we have proposed a solution to the annoying problem of building the topology of a Bayesian network by filtering the relevant attributes using a PSO-based feature selection approach. Once the reduct has been computed, a Bayesian network is built from scratch by means of the K2 algorithm. Our contribution lies in the demonstration of the direct concordance between the values of the fitness function associated to the best reduct and the rest of the traditionally used validation indicators for classification, therefore guessing in advance the performance of the Bayesian network during classification from the fitness value achieved for computing its best reduct.

As a future work we will focus on several specific scenarios where the so-built Bayesian network outperforms traditional approaches.

**Acknowledgments.** This work was developed in the framework of a collaboration program supported by VLIR (Flemish Interuniversity Council, Belgium). We would also like to thank the critical and helpful comments and suggestions of the referees so as to improve the paper's readability and overall quality.

## References

1. Baldi, P., Brunak, S.: Assessing the accuracy of prediction Algorithms for classification: An Overview. *Bioinformatics* 16(5), 412–424 (2000)
2. Bockhorst, J., Craven, M., Page, D., Shavlik, J., Glasner, J.: A Bayesian network approach to operon prediction. *Bioinformatics* 19(10), 1471227–1471235 (2003)
3. Bouckaert, R.R.: Bayesian Network Classifiers in Weka (2004)
4. Brazma, A., Jonassen, I.: Context - specific independence in Bayesian networks. In: Proc. Twelfth Conference on Uncertainty in Artificial Intelligence, pp. 115–123 (1996)
5. Grau, R., Galpert, D., Chávez, M., Sánchez, R., Casas, G., Morgado, E.: Algunas aplicaciones de la estructura booleana del Código Genético. *Revista Cubana de Ciencias Informáticas*, Año 1, vol 1 (2005)
6. Hunter, L.: Artificial Intelligence and Molecular Biology. p. 500, references, index, illus. electronic text, <http://www.biosino.org/mirror/www.aaai.org/Press/Books/Hunter/default.htm> ISBN 0-262-58115-9
7. Marchal, K., Thijs, G., De Keersmaecker, S., Monsieurs, P., De Moor, B., Vanderleyden, J.: Genome-specific higher-order background models to improve motif detection. *Trends in Microbiology* 11(2), 61–66 (2003)
8. Mahamed, G.H.O., Andries, P.E., Ayed, S.: Dynamic Clustering using PSO with Application in Unsupervised Image Classification. *Transactions on Engineering, computing and Technology* 9 (2005)
9. Kennedy, J.: The particle swarm: social adaptation of knowledge. In: IEEE International Conference on Evolutionary Computation, April 13–16, 1997, pp. 303–308 (1997)
10. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Computer Society Press, Los Alamitos (1995)
11. Kennedy, J., Eberhart, R.C.: A new optimizer using particle swarm theory. In: Sixth International Symposium on Micro Machine and Human Science, Nagoya, pp. 39–43 (1995)
12. Kennedy, J., Spears, W.M.: Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In: Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 39–43. IEEE Computer Society Press, Los Alamitos (1998)
13. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition* 33(1), 25–41 (2000)
14. Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J.A., Armañanzas, R., Santafé, G., Pérez, G., Robles, V.: Machine learning in bioinformatics. *BIOINFORMATICS* 7(1), 86–112 (2005)
15. Liu, H., Li, J., Wong, L.: A Comparative Study on Feature Selection and Classification Methods Using Gene Expression Profiles and Proteomic Patterns. *Genome Informatics* 13, 51–60 (2002)
16. Liu, H., Setiono, R.: Chi2: Feature selection and discretization of numeric attributes. In: Proc. IEEE 7th International Conference on Tools with Artificial Intelligence, pp. 338–391. IEEE Computer Society Press, Los Alamitos (1995)
17. Mellors, J.W., Brendan, A.L., Schinazi, R.F.: Mutations in HIV-1 Reverse Transcriptase and Protease Associated with Drug Resistance

18. Murray, R.J.: Predicting Human Immunodeficiency Virus Type 1 Drug Resistance From Genotype Using Machine Learning. Master of Science School of Informatics. University of Edinburgh (2004)
19. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishing, Dordrecht (1991)
20. Sánchez, R., Grau, R., Morgado, E.: Genetic Code Boolean Algebras. *WSEAS Transactions on Biology and Biomedicine* 1(2), 190–197 (2004)
21. Sánchez, R., Morgado, E., Grau, R.: A genetic code boolean structure I. The meaning of boolean deductions. *Bulletin of Mathematical Biology* 67, 1–14 (2005)
22. Sánchez, R., Morgado, E., Grau, R.: The genetic code boolean lattice, *MATCH Communications in Mathematical and in Computer Chemistry*, vol. 52, pp. 29–46 (2004)
23. Siegel, S.: *Diseño Experimental no parametrico*, segunda edicion, 245–256 (1987)
24. Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R.: Feature Selection Based on Rough Sets and Particle Swarm Optimization. In: *Pattern Recognition Letter*, Elsevier, Amsterdam (2006)
25. Witten, I.H., Frank, E.: *Data Mining Practical Machine Learning Tools and Techniques*, pp. 363–483. Morgan Kaufman, San Francisco (2005)

# Two Simple and Effective Feature Selection Methods for Continuous Attributes with Discrete Multi-class

Manuel Mejía-Lavalle, Eduardo F. Morales, and Gustavo Arroyo

Instituto de Investigaciones Eléctricas, Reforma 113, 62490 Cuernavaca, Morelos, México  
INAOE, L.E. Erro 1, 72840 StMa. Tonantzintla, Puebla, México  
mlavalle@iie.org.mx, emorales@inaoep.mx, garroyo@iie.org.mx

**Abstract.** We present two feature selection methods, inspired in the Shannon's entropy and the Information Gain measures, that are easy to implement. These methods apply when we have a database with continuous attributes and discrete multi-class. The first method applies when attributes are independent among them given the class. The second method is useful when we suspect that interdependencies among the attributes exist. In the experiments that we realized, with synthetic and real databases, the proposed methods are shown to be fast and to produce near optimum solutions, with a good feature reduction ratio.

## 1 Introduction

Feature selection has shown to be promising pre-processing step in data mining because it can eliminate the irrelevant or redundant attributes that cause the mining tools to become inefficient and ineffective [1]; at the same time, it can preserve, and in many cases, increase the classification quality of the mining algorithm and help in the understanding of the induced models, as they tend to be smaller.

Although there are many feature selection algorithms reported in the specialized literature [1], none of them is perfect: some of them are effective, but very costly in computational time (e.g., *wrappers* methods [2]), and others are fast, but less effective in the feature selection task (e.g., *filter* methods [3]). Most of them need pure discrete data (e.g., nominal attributes with a nominal class) or pure continuous data (e.g., continuous attributes with a continuous class). So, if data has continuous attributes, they need to be discretized (either all the attributes or the class), however the results vary depending on the discretization method that is utilized [4].

In this article we propose two easy to implement feature selection methods that apply over continuous data with discrete class in a supervised learning context. The first method assumes that the attributes are independent among them given the class. The second method is useful when we suspect that interdependencies among the attributes exist. Both methods are inspired in the Shannon's  $n$ -dimensional entropy and the Information Gain measures. We show that the proposed methods are fast and produce near optimum solutions, selecting few attributes, according to the experiments that we realized, with synthetic and real databases.

To cover these topics, the article is organized as follows: Section 2 introduces our feature selection methods; Section 3 details the experiments; in Section 4 we discuss



and survey some works related with ours methods; conclusions and future research directions are offered in Section 5.

## 2 Proposed Feature Selection Methods

### 2.1 vG Method: When Attributes Are Independent

vG is inspired in Shannon’s entropy and Information Gain, which emerged from the Information Theory arena. So, we begin our description introducing these basic concepts [5]. Formally, the entropy  $H_n$  of a nominal, or discrete, set of probabilities  $p_1, \dots, p_n$  has been defined as:

$$H_n = - \sum p_i \log_2 p_i \tag{1}$$

Additionally, there is a less used, and known, entropy version  $H_c$  for continuous data; according to [5] is defined as:

$$H_c = - \int p(x) \log_2 p(x) dx \tag{2}$$

but generally, the density distribution function  $p(x)$ , is unknown: Miller[6] tried to estimate this function, using Voronoi regions (a kind of discretization), but he concluded that this process has exponential complexity). As Shannon [5], physicists and statisticians point out [6], a reasonable approach is to assume that this density distribution is Gaussian, whose standard deviation is  $S$ . If we realize some algebraic manipulations over (2), assuming  $p(x)$  to be Gaussian, we obtain:

$$H_c(x) = \log_2 \{ 2 \pi e \}^{1/2} S \tag{3}$$

Observing (3) we can say that, in this terms, the entropy of one-dimensional Gaussian distribution depends on its standard deviation  $S$ : if  $S$  is relative small, then the entropy is small, and vice versa.

On the other hand, *Information Gain* (over nominal data, with nominal classes  $c_1, \dots, c_n$ ) tell us how much information we obtain if consider some particular attribute:

$$I_n(x) = H_n(c_1, \dots, c_n) - \{ Q_1/T H_n(c_1) + \dots + Q_n/T H_n(c_n) \} \tag{4}$$

where  $Q_n/T$  is the weight (instances quantity) for class  $n$ , respect to the total instance quantity  $T$ .

So, if we combine these ideas, we obtain a new form of Information Gain  $I_c$  applied to continuous data. For a database with continuous attributes and  $n$  nominal classes we propose the next equation (where  $S^2$  means variance):

$$I_c(x) = S^2(c_1, \dots, c_n) - \{ (Q_1/T) S^2(c_1) + \dots + (Q_n/T) S^2(c_n) \} \tag{5}$$

With equation (5) we can obtain feature relevance in a filter-ranking fashion, without requiring parameters' adjustments. For example, if we have a continue attribute At1:

At1 Values	0.8	0.7	0.8	0.6	0.2	0.1	0.3	0.1	0.3
At1 Class	$c_1$	$C_1$	$C_1$	$c_1$	$c_2$	$C_2$	$c_2$	$c_2$	$c_2$

Then  $S^2(c_1, \dots, c_n) = S^2(0.8,0.7,0.8,0.6,0.2,0.1,0.3,0.1,0.3) = 0.085$ , and too  $S^2(c_1) = S^2(0.8,0.7,0.8,0.6) = 0.009$ , and  $S^2(c_2) = S^2(0.2,0.1,0.3,0.1,0.3) = 0.01$ .

Applying equation (5):  $I_c (At1) = 0.085 - \{ 4/9 * 0.009 + 5/9 * 0.01 \} = 0.075$ . If we repeat the process for more attributes and we obtain that:  $I_c (At2) = 0.003$ ,  $I_c (At3) = 0.28$ ,  $I_c (At4) = 0.04$ , then the attribute ranking is At3, At1, At4, At2, where At3 is the best attribute, and so on.

By analogy, we call our method for feature selection as *Variance Gain* ( $vG$ ). Thus, the proposed method consist of:

1. Perform data normalization<sup>1</sup>, between 0 and 1 (to maintain the same scale for all database continuous attributes).
2. Apply  $vG$  to each attribute (to obtain the relevance for each one).
3. Realize a descending ordering attribute-metric (attribute ranking process).
4. Select the best attributes (to select the best ranking attributes, we use a threshold defined by the largest gap between two consecutive ranked attributes, e.g., a gap greater than the average gap among all the gaps, according to [4]).
5. Use the selected attributes to perform induction (data mining process).

$vG$  uses a simple metric based on testing decreasing values of variance in the class after selecting an attribute and results useful for the feature selection task. As shown in Section 3,  $vG$  is also a very effective and competitive alternative.

## 2.2 $dG$ Method: When There Are Interdependencies Among the Attributes

The proposed method to realize non-myopic [5] feature selection is inspired also in the Shannon’s (n-dimensional) entropy and the Information Gain measures. So, in analogous way, we begin our description introducing related basic concepts.

Formally, the entropy  $H$  of a numerical, or continuous distribution with an n-dimensional distribution  $p(x_1, x_2, \dots, x_n)$  has been defined [6] as:

$$H = - \int \dots \int p(x_1, x_2, \dots, x_n) \log_2 p(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n \tag{6}$$

Generally  $p(x_1, x_2, \dots, x_n)$  is unknown, and must be estimated, but this process has exponential complexity [7]. Again, following to Shannon, physicists and statisticians, we assume that this density distribution is Gaussian. If we realize some algebraic manipulations over (6), assuming the n-dimensional Gaussian distribution with associated quadratic form  $a_{ij}$  we obtain:

$$H = \log_2 \{ 2 \pi e \}^{n/2} |a_{ij}|^{-1/2} \tag{7}$$

where  $|a_{ij}|$  is the determinant whose elements are the covariance matrix  $a_{ij}$ .

Observing (7) we can say that, for this case, the n-dimensional entropy depends on the covariance matrix determinant. Indeed, the covariance matrix is the natural generalization to higher dimensions of the concept of the variance of a scalar-valued random variable. Also we point out that the geometric meaning of a determinant is just the volume of the n-dimensional parallelepiped that  $n$  vectors (in our case,  $n$  attributes) forms. This means that, while more volume, the vectors are more independents and therefore, a larger n-dimensional entropy is obtained, and vice-versa: if  $|a_{ij}|$  is relative small, then the n-dimensional entropy is small, indicating us that the features are inter-dependents.

---

<sup>1</sup> One can easily scale it so the variance is 1 and mean is 0, which is also popular.

So again, we propose combine ideas from equations (4) and (7) to obtain a new form of a non-myopic Information Gain:  $dG$  (determinant Gain) applied to continuous and  $n$ -dimensional data. Then, for a database with continuous attributes and discrete multi-class we propose the next equation:

$$dG = |a_{ij}|(c_1, \dots, c_n) - \{ (Q_1/T) |a_{ij}|(c_1) + \dots + (Q_n/T) |a_{ij}|(c_n) \} \quad (8)$$

With equation (8) we can obtain a measure of subset feature relevance, without tuning of parameters and in a very simple way. For instance, we can consider the well-known XOR problem: applying the proposed metric we obtain the following evaluations for attributes  $X$  and  $Y$ :  $dG(X) = 0 - (0 + 0) = 0.0$ ;  $dG(Y) = 0 - (0 + 0) = 0.0$ ;  $dG(X, Y) = 0.0625 - (0 + 0) = 0.0625$ .

These results imply that if we consider  $X$  or  $Y$  in an isolated fashion, they cannot predict the class. On the other hand, if we evaluated the joint contribution of  $X$  and  $Y$ , the  $dG$  value is greater, and therefore implies that this subset is a better predictor of the class. In order to find the best (or near best) attribute subset we can apply *best-first search*. Thus, the non-myopic feature selection method proposed consist of:

Given a dataset with  $N$  attributes and  $M$  instances (where  $\| \cdot \|$  is the cardinal of a set):

1. Perform data normalization, between 0 and 1 (to maintain the same scale for all database continuous attributes);
2. Apply  $dG$  for each attribute;
3. While (available memory) or (unexplored nodes) do  
begin  
select for expansion the feature subset  $F$  with the best  $dG$   
(and better than his parent node);  
for  $I := 1$  to  $(N - \|F\|)$  do  
begin  
obtain  $dG(F \cup I | I \notin F)$ ;  
end;  
end;
4. Use the best evaluated attribute subset to perform induction (data mining process).

The proposed method  $dG$  is tested in the next Section.

### 3 Experiments

We conducted several experiments with real and synthetic datasets to empirically evaluate if  $vG$  and  $dG$  can do better in selecting features than other well-known feature selection algorithms, in terms of learning accuracy, attribute reduction and processing time. We also choose synthetic datasets in our experiments because the relevant features of these datasets are known beforehand.

#### 3.1 Experimentation Details

The experimentation objective is to observe  $vG$  and  $dG$  behavior related to classification quality (predictive accuracy), attribute reduction and response time.

First, we test ours proposed methods with a real database with 24 attributes and 35,983 instances; this database contains information of Mexican electric billing customers, where we expect to obtain patterns of behavior of illicit customers.

We test too with five well-known databases, taken form the UCI repository [8] (see Table 2 for details).

To obtain additional evidence, we experiment with the corrAL (and corrAL-47) synthetic dataset, proposed in [4], that has four relevant attributes (A0, A1, B0, B1), plus one irrelevant ( I ) and one redundant ( R ) attributes; the class attribute is then defined by the function  $Y = (A0 \wedge A1) \vee (B0 \wedge B1)$ .

In order to compare the results obtained with  $vG$  and  $dG$ , we use Weka's [9] implementation of ReliefF, CFS, OneR and ChiSquared feature selection algorithms. These implementations were run using Weka's default values, except for ReliefF, where we define to 5 the number of neighborhood, for a more efficient response time. Additionally, we experiment with 7 Elvira's [10] filter-ranking methods: Bhattacharyya, Matusita, Euclidean, Mutual Information, Shannon entropy, Kullback-Leibler 1 and 2. All the experiments were executed in a personal computer with a Pentium 4 processor, 1.5 GHz, and 250 Mbytes in RAM. In the following Section the obtained results are shown.

**Table 1.** J4.8's accuracies for 10-fold-cross validation using the features selected by each method (Electric billing database)

Method	Total features selected	Accuracy (%)	Pre-processing time
CFS	1	90.18	9 secs.
<i>dG</i>	2	90.70	43 secs.
<i>vG</i>	3	94.02	0.7 secs.
Bhattacharyya	3	90.21	6 secs.
Matusita distance	3	90.21	5 secs.
ReliefF	4	93.89	14.3 mins.
Euclidean distance	4	93.89	5 secs.
Kullback-Leibler 1	4	90.10	6 secs.
Mutual Information	4	90.10	4 secs.
Kullback-Leibler 2	9	97.50	6 secs.
OneR	9	95.95	41 secs.
Shannon entropy	18	93.71	4 secs.
ChiSquared	20	97.18	9 secs.
All attributes	24	97.25	0

### 3.2 Experimental Results

Testing over the Mexican electric billing database, we use the selected features for each method as input to the decision tree induction algorithm J4.8 included in the Weka tool (J4.8 is the last version of C4.5, which is one of the best-known induction algorithms used in data mining). We notice that  $dG$  obtains good accuracy with only 2 attributes, better than other methods that select 3 and 4 attributes (Table 1). On the other hand,  $dG$  is faster than ReliefF, although this method obtains better accuracy, but selecting more attributes (4 attributes). We notice too that  $vG$  obtains an excellent accuracy with 3 features and it has the best processing time.

To have a better idea of the  $vG$  and  $dG$  performance, we can compare the results presented previously against the results produced by an exhaustive wrapper approach. In this case, we can calculate that, if the average time required to obtain a tree using J4.8 is 1.1 seconds, and if we multiply this by all the possible attribute combinations, then we will obtain that 12.5 days, theoretically, would be required to conclude such a process.

Testing over five UCI datasets,  $vG$  and  $dG$  obtains similar average accuracy as CFS and ReliefF, but in general with less processing time and better feature reduction than ReliefF (Table 2). SOAP's results were taken from [11]: although this method is very fast, it cannot reduce considerably the quantity of attributes for Ionosphere dataset.

**Table 2.** J4.8's accuracies using the features selected by each method for five UCI datasets

Method	Autos (25/205/7)			Horse-c (27/368/2)			Hypothyroid (29/3772/4)			Sonar (60/208/2)			Ionosphere (34/351/2)			Avg. Acc
	TF	Ac	Pt	TF	Ac	Pt	TF	Ac	Pt	TF	Ac	Pt	TF	Ac	Pt	
All atts	25	82	0	27	66	0	29	99	0	60	74	0	34	91	0	<b>82.4</b>
$vG$	8	75	0.01	3	69	0.02	4	95	0.2	11	73	0.03	4	91	0.3	<b>80.6</b>
$dG$	7	75	12	2	68	14	5	95	26	9	75	14	3	88	18	<b>80.2</b>
CFS	6	74	0.05	2	66	0.04	2	96	0.3	18	74	0.09	8	90	3	<b>80.0</b>
ReliefF	11	74	0.4	3	66	0.9	6	93	95	4	70	0.9	6	93	4	<b>79.2</b>
SOAP	3	73	0.01	3	66	0.02	2	95	0.2	3	70	0.02	31	90	0.01	<b>78.8</b>
Mutual I	3	72	0.9	4	68	1	2	90	1.4	18	73	1	3	86	1	<b>77.8</b>
OneR	5	70	0.8	3	67	1	3	88	1.3	12	72	1	4	85	1	<b>76.4</b>
KL-1	3	71	0.9	4	61	1.2	3	92	1.7	16	70	1	2	86	1	<b>76.0</b>
KL-2	4	68	0.9	4	62	1.1	2	89	1.5	11	68	1	3	83	1	<b>74.0</b>
Matusita	3	66	1.7	3	61	2.3	2	91	3.3	17	68	2.5	2	83	2	<b>73.8</b>
Bhattach	3	67	0.8	3	60	1	1	90	1.4	9	68	1	2	83	1	<b>73.6</b>
Euclidean	2	66	1	3	62	1.4	2	90	1.2	10	67	1.1	2	82	1	<b>73.4</b>
ChiSqua	3	67	1	2	60	1.6	3	88	1.3	11	65	1.2	2	80	1	<b>72.0</b>
Shannon	4	66	0.9	4	61	1.3	2	87	1.6	9	66	1	2	80	1	<b>72.0</b>

“(25/205/7)” means (attributes, instances, classes) for Autos dataset, and so on.

TF=Total features selected

Ac=Accuracy (%)

Pt=Pre-processing time (secs.)

Finally, when we test with the corrAL and corrAL-47 datasets [4], our  $dG$  method produces the best results (Table 3) because it selects the perfect attributes, it is to say, it can detect effectively the important ones (A0, A1, B0 and B1); although  $vG$  cannot obtain the perfect quantity of attributes, it can detect the important ones; results for FCBF, CFS and Focus methods was taken from [4]. Elvira's ranking methods obtain poor results, so we prefer instead show results for Symmetrical Uncertainty (SU) and Gain Ratio metrics.

**Table 3.** Features selected by different methods (corrAL and corrAL-47 datasets)

Method	CorrAL	corrAL-47
$dG$	A0, A1, B0, B1	A0, A1, B0, B1
$vG$	R, A0, A1, B0, B1	R,B1,B1 <sub>1</sub> ,A1,A1 <sub>1</sub> ,A0,A0 <sub>1</sub> ,B0
ReliefF	R, A0, A1, B0, B1	R,B1 <sub>1</sub> ,A0,A0 <sub>0</sub> ,B1,B1 <sub>0</sub> ,B0,B0 <sub>0</sub> ,B0 <sub>2</sub> ,A1,A1 <sub>0</sub>
FCBF <sub>(log)</sub>	R, A0	R, A0, A1, B0, B1
FCBF <sub>(0)</sub>	R, A0, A1, B0, B1	R, A0, A1, B0, B1
CFS	A0, A1, B0, B1, R	A0, A1, B0, B1, R
Focus	R	A0, A1, A1 <sub>2</sub> , B0, B1, R
SU (Weka)	R, A1, A0, B0, B1	A0 <sub>1</sub> , A0, A0 <sub>7</sub> , B0 <sub>1</sub> , B0, A1 <sub>1</sub> , A1, R
Gain Ratio (Weka)	R, A1, A0, B0, B1	A0 <sub>1</sub> ,A0,A0 <sub>7</sub> ,B0,B0 <sub>1</sub> , A1, R, A1 <sub>1</sub>
OneR	R, A1, A0, B0, B1	A0 <sub>1</sub> ,A0,A0 <sub>7</sub> ,B0 <sub>1</sub> ,B0, A1 <sub>1</sub> , A1, R, A0 <sub>5</sub> , B1 <sub>3</sub>
ChiSquared	R, A1, A0, B0, B1	A0 <sub>1</sub> ,A0,A0 <sub>7</sub> , B0 <sub>1</sub> ,B0, A1 <sub>1</sub> , R, A1, B1 <sub>3</sub>

We point out that these results suggest that  $dG$  method effectively captures inter-dependencies among attributes and therefore, it is a non-myopic feature selection method.

Pre-processing time for  $dG$  method was inferior to one second when experiment with the corrAL dataset, and with the corrAL-47 dataset this time was 59 seconds (due to combinatorial search): we think that this response time is reasonable, but we recognize that the other ten tested methods only need around 10 seconds to conclude this task.

## 4 Discussion and Related Work

There is a great variety of feature selection filter methods for nominal data. Some authors consider the ID3 algorithm [12] as one of the first proposed approaches to filter (in a embedded way). Although some ID3's extensions (like C4.5 and J4.8) manages continuous data, they perform a kind of internal binary split discretization so, these extensions, do not operate directly over continuous data.

Among the pioneering filter methods, and very much cited, are Focus [13] (that makes an exhaustive search of all the possible attribute subsets, but this is only appropriate for problems with few attributes), and Relief [14] and ReliefF (that has the disadvantage of not being able to detect redundant attributes, and also it is time consuming).

Koller [15] uses a distance metric called cross-entropy or KL-distance, that compares two probability distributions and indicates the error, or distances, among them, plus a Markov Blanket, and obtains around 50% reduction on the number of attributes, maintaining the quality of classification and being able to significantly reduce processing times (for example, from 15 hours of a wrapper scheme application, to 15 minutes for the proposed algorithm). The final result is “sub optimal” because it assumes independence between attributes, which it is not always true.

Piramuthu [3] evaluates 10 different measures for the attribute-class distance, using Sequential Forward Search (SFS), that includes the best attributes selected by each measure into a subset, such that the final result is a better attribute subset than the individual groups proposed by each method. However, the results are not compared with the original complete attribute set and so, it is not possible to conclude anything about the effectiveness of each measure; although SFS manages to reduce the search space, multiple mining algorithm runs, varying the attribute subsets, are necessary to validate the scheme and this is computationally expensive.

SOAP is a method that operates on numerical attributes and discrete or nominal class [11] and has a low computational cost: it counts the number of times the class value changes with respect to an attribute whose values have been sorted into ascending order. SOAP reduces the number of attributes as compared to other methods; nevertheless, the user has to supply the number of attributes that will be used in the final subset. This is a common problem with the *filter-ranking* methods, that output a ordered list of all attributes, according to its relevance.

In the scenario with pure continuous data, we can apply Regression Tress [16]: this method determines relevant attributes by means of co-variance between each continuous attribute and the continuous class.

Molina [17] tried to characterize 10 different feature selection methods by measuring the impact of redundant and irrelevant attributes, as well as of the number of instances. Significant differences could not be obtained, and it was observed that, in general, the results of the different methods depended on the data being used.

Perner and Apté [5] realized an empirical evaluation of feature selection based on a real-world data set, applying the CM feature subset selection method: they showed that accuracy of the C4.5 classifier could be improved with an appropriate feature pre-selection phase that at the same time reduces attribute quantity; however, due to the paper’s goal, they did not realized further experiments to emphasize the CM’s response time, attribute reduction or the CM’s ability to detect attribute interactions.

Other proposals for feature selection explore the use of neural networks, fuzzy logic, genetic algorithms, and support vector machines [1], but they are computationally expensive and have one, or more, user’s parameters to adjust. In general, it is observed that the methods that have been proposed: a) need nominal data; b) obtain results that vary with the domain of the application; c) obtain greater quality results, only with greater computational cost; d) depend on suitable tuning; and e) they suffer of myopic feature selection.

## 5 Conclusions and Future Work

We have presented two feature selection new methods easy to implement that try to overcome some drawbacks found with traditional pure nominal or discrete feature selection methods.

From the experimentations presented, with a real Mexican electric billing database, five UCI datasets and two synthetic datasets, the proposed methods  $vG$  and  $dG$  represents a promising alternatives, compared to other methods, because of its acceptable processing time and good performance in the feature selection task in both, accuracy and attribute reduction. Additionally, ours methods works without user parameters that generally imply some kind of special and time consuming tuning.

Some future research issues arise with respect to  $vG$  and  $dG$  testing and improvement. For example: experimenting with more real and challenging databases (e.g., future work will be the application of the formalism to other very large power system databases such as the national power generation performance database, the national transmission energy control databases, the de-regulated energy market database, and the Mexican electric energy distribution database); applying other data mining induction-classification algorithms (e.g., Naïve Bayes classifier, 1NN, etc.); perform more experiments following [18]; apply statistical tests to observe if the differences in accuracies of the proposed methods are significant and apply other functions to overcome the Gaussian distribution assumption.

## References

1. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of machine learning research* 3, 1157–1182 (2003)
2. Kohavi, R., John, G.: Wrappers for feature subset selection. *Artificial Intelligence Journal*, Special issue on relevance, 273–324 (1997)
3. Piramuthu, S.: Evaluating feature selection methods for learning in data mining applications. In: *Proc. 31st annual Hawaii Int. conf. on system sciences*, pp. 294–301 (1998)
4. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research* 5, 1205–1224 (2004)
5. Perner, P., Apté, C.: Empirical Evaluation of Feature Subset Selection Based on a Real-World Data Set. In: Zighed, A.D.A., Komorowski, J., Żytkow, J.M. (eds.) *PKDD 2000. LNCS (LNAI)*, vol. 1910, pp. 575–580. Springer, Heidelberg (2000)
6. Shannon, C.E.: A mathematical theory of communication. *Bell System Technical Journal* 27, 379–423, 623–656 (1948)
7. Miller, E.: A new class of entropy estimators for multi-dimensional densities. In: *Int.conference on Acoustics, Speech and Signal Processing* (2003)
8. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: *UCI Repository of machine learning databases*, Department of Information and Computer Science. University of California, Irvine, CA (1998), <http://www.ics.uci.edu/mllearn/MLRepository.html>
9. [www.cs.waikato.ac.nz/ml/weka](http://www.cs.waikato.ac.nz/ml/weka) (2004)
10. [www.ia.uned.es/elvira/](http://www.ia.uned.es/elvira/) (2004)
11. Ruiz, R., Aguilar, J., Riquelme, J., SOAP: efficient feature selection of numeric attributes, VIII Iberamia, workshop de minería de datos y aprendizaje, Spain, 2002, pp. 233–242.
12. Quinlan, J.: Unknown attribute values in ID3. In: *Int. conf. Machine learning*, pp. 164–168 (1989)
13. Almuallim, H., Dietterich, T.: Learning with many irrelevant features. In: *Ninth nat. conf. on AI*, pp. 547–552. MIT Press, Cambridge (1991)
14. Kira, K., Rendell, L.: The feature selection problem: traditional methods and a new algorithm. In: *Tenth nat. conf. on AI*, pp. 129–134. MIT Press, Cambridge (1992)



15. Koller, D., Sahami, M.: Toward optimal feature selection. In: Int. conf. on machine learning, pp. 284–292 (1996)
16. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth International Group, Belmont, CA (1984)
17. Molina, L., Belanche, L., Nebot, A.: Feature selection algorithms, a survey and experimental eval. In: IEEE Int.conf.data mining, Maebashi City Japan, pp. 306–313. IEEE Computer Society Press, Los Alamitos (2002)
18. Ambrose, C., McLachlan, G.J.: Selection Bias in Gene Extraction in Tumour Classification on Basis of Microarray Gene Expression Data. PNAS 99(10), 6562–6566 (2002)

# INCRAIN: An Incremental Approach for the Gravitational Clustering

Jonatan Gomez, Juan Peña-Kaltekis, Nestor Romero-Leon, and Elizabeth Leon

Universidad Nacional de Colombia

Computer Systems Engineering Department

{jgomezpe, eleonguz}@unal.edu.co, {juanerp, romero.nestor}@gmail.com

**Abstract.** This paper introduces an incremental data clustering algorithm based on the gravitational law. Basically, data samples are considered as unit-mass particles exposed to gravitational forces. Data points are clustered according their proximity during the simulation of the dynamical system defined by their gravitational fields. When the simulation is stopped, a set of prototypes is generated (several prototypes per cluster found). Each prototype will have associated a mass that is proportional to the number of particles in the sub-cluster and will be used as additional particle when new data samples are given for clustering. Experiments are performed on synthetic data sets and the obtained results are presented.

## 1 Introduction

Clustering is an unsupervised learning technique that organizes data samples into separated groups or clusters, without any previous knowledge about the class or group such samples belong to [9]. Groups or Clusters are defined in such a way that data points assigned to the same cluster have a high similarity between them, while data points assigned to different clusters have a low similarity between them. Although there are many different approaches for data clustering, such techniques can be classified in four broad categories: Partitioning, Hierarchical, Density and Hybrid based methods [9]. A partitioning algorithm produces a single partition of the data set by applying a criterion function (usually the squared error criterion) [4]. Well known partitioning techniques are K-Means, K-Medians, and CLARA [4,3]. A hierarchical algorithm produces a dendrogram presenting a nested grouping of data set and similarity levels at which the clusters change. Most hierarchical clustering algorithms are variants of the single link, the complete link and minimum variance algorithms [9]. In particular, the agglomerative method or bottom-up, starts with a cluster for every data sample and joins the most similar groups until only a single groups remains or until some stop criterion is fulfilled (e.g., AGNES [1], IHC [13]). In density based clustering algorithms, data samples are grouped while their density (number of samples) in the vicinity satisfies certain threshold or while new samples are accessible or connected to each other through the vicinities of the core samples belonging to the identified clusters. (e.g., DBSCAN [4,12], DENCLUE). Another approaches

use grids for dividing the data set space (e.g STING [12], CLIQUE), or build statistical models that establish the best fit for the clusters in the data set (e.g COBWEB, AutoClass [11,5]) or use gravity force as a mechanism for grouping data points and identifying clusters [7].

Although many of these techniques are widely used, all of them face the same challenge when it comes to reduce the computational cost, specially in memory, of using them in large data sets. Incremental clustering approaches try to analyze large data sets in environments with limited computational resources [6]. The main idea of incremental clustering techniques is to work with a partition of the data set and some information about the previously analyzed data set partitions.

The purpose of this paper is to develop an incremental version of the RAIN algorithm [8] based on the mass concept and its effect in space. The mass concept allows the algorithm to represent several data points belonging to a cluster in a single point (prototype). This paper is divided in 6 sections: Section 2 gives an overview of incremental clustering; Section 3 covers the main ideas behind the RAIN algorithm; Section 4 introduces the new approach called INCRAIN; Section 5 presents the performed experiments and the analysis of the results, and Section 6 draws some conclusions and discusses some future work.

## 2 Incremental Clustering

The incremental clustering problem can be defined as follows: Given a set of  $n$  data points, in a metric or quasi metric space  $M$ , maintain a collection of clusters in such a way that if new data points are observed one or all of the following situations happens: a) they are assigned to such clusters b) New clusters are generated. c) Some clusters are merged [6,10,11].

According to this definition, there are two main characteristics of incremental clustering techniques:

1. Incremental clustering techniques have the capability of working with dynamic data sets that does not even exist. In this way, an incremental clustering technique becomes necessary when the cost of constant data re-clustering surpasses the benefit of a non incremental clustering technique.
2. Incremental clustering algorithms face the challenge of diminish the uncertainty generated by analyzing the data set in parts (not in full). In this way, for incremental clustering techniques is very important to determine the way data and knowledge is represented [6].

Although there are several incremental clustering techniques, there are two main approaches: In one side there is the single observation approach and in the other side there is the divide and conquer approach.

The single observation aims for analyzing a data point when it just arrives. Then, two decisions can be made: a) a new cluster is generated or b) the data point is assigned to an existing cluster. Several variations of this mechanism maintain the data points in their clusters representation (with the consequent growing of the space required for maintaining such representation), while others develop prototypes to represent the analyzed data.

---

**Algorithm 1.** Randomized Gravitational Clustering

---

```

Rgc(  $x$ ,  $G$ ,  $\Delta(G)$ ,  $M$ ,  $\varepsilon$ )
1  for  $i=1$  to  $N$  do
2    MAKE( $i$ )
3  for  $i=1$  to  $M$  do
4    for  $j=1$  to  $N$  do
5       $k$  = random point index such that  $k \neq j$ 
6      MOVE(  $x_j$ ,  $x_k$  ) //Move both points
7      if  $\text{dist}(x_j, x_k) \leq \varepsilon$  then UNION(  $j$ ,  $k$  )
8       $G = (1-\Delta(G))*G$ 
9  for  $i=1$  to  $N$  do
10   FIND( $i$ )
11  return disjoint-sets

```

---

The divide and conquer approach creates a partition of the data set rather than using a single point each time. For each partition of the data set, a non-incremental clustering algorithm is used to analyze the data set (the previously generated information and the current partition). It is clear that techniques within this approach must generate representative prototypes of the data (previously generated prototypes and data samples in the partition) being used during the iteration as a starting point for the next iteration.

### 3 Rain: Data Clustering Using Randomized Interaction of Data Points

In [7], Gomez et al developed a robust clustering technique based on the gravitational law and Newton’s second motion law. In this way, for an  $n$ -dimensional data set with  $N$  data points, each data point is considered as a particle in the  $n$ -dimensional space with mass equal to 1. Each particle is moved according to a simplified and generalized version of the Gravitational Law using the Second Newton’s Motion Law. Algorithm 1 shows the randomized gravitational clustering algorithm.

Function MOVE (line 6), moves both points  $x_j$  and  $x_k$  using a generalized and simplified version of the gravitational law movement function, see equation 1.

$$x(t + 1) = x(t) + \vec{d} \frac{G}{\|\vec{d}\|^3} \tag{1}$$

here,  $\vec{d} = \vec{y} - \vec{x}$ , and  $G$  is the gravitational constant. In order to eliminate the big crunch limit effect, the gravitational constant  $G$  is reduced each iteration in a constant proportion (the decay term:  $\Delta(G)$ ). RGC creates a set of clusters by using an optimal disjoint set union-find structure [2]. When two points are merged, both of them are kept in the system while the associated set structure is modified. In order to determine the new position of each data point, the proposed

---

**Algorithm 2.** Cluster Extraction

---

```

GetClusters( clusters,  $\alpha$ ,  $N$  )
1 newClusters =  $\emptyset$ 
2 MIN_POINTS =  $\alpha N$ 
3 for i=0 to number of clusters do
4   if size( clusteri )  $\geq$  MIN_POINTS then
5     newClusters = newClusters  $\cup$  { clusteri }
6 return newClusters
    
```

---

algorithm only selects another data point in a random way and move both of them according to equation 1 (MOVE function). RGC returns the sets stored in the disjoint set union-find structure. Because RGC assigns every point in the data set (noisy or normal) to one cluster, it is necessary to extract the valid clusters. RGC uses an extra parameter ( $\alpha$ ) to determine the minimum number of points (percentage of the training data set) that a cluster should include in order to be considered a valid cluster , see Algorithm 2.

RAIN extends the RGC algorithm in such a way that different decreasing functions can be used instead of the one based on the Gravitational Law [8]. In order to reduce the sensitivity of RAIN to the size of the data set, a rough estimate of the maximum distance between closest points in the data set is estimated, see equation 2. Given a collection of  $N$  data points in the  $n$ -dimensional  $[0, 1]$  Euclidean space, the maximum distance between closest points can be roughly approximate using equation .

$$\hat{d} = \frac{2 * \sqrt{n}}{\sqrt{3} * N^{\frac{1}{n}}} \tag{2}$$

Although motivated by the Gravitational Law and second Newton’s motion law, RGC can be seen as an algorithm that moves interacting data points according to a decreasing function of the data points distance. In RAIN, the final position of a data point  $x$  that is interacting with another data point  $y$  is defined by equation 3.

$$x(t + 1) = x(t) + G * \vec{d} * f \left( \frac{\|\vec{d}\|}{\hat{d}} \right) \tag{3}$$

where,  $\vec{d} = \vec{y} - \vec{x}$ ,  $f$  is a decreasing function,  $\hat{d}$  is the rough estimate of maximum distance between closest points and  $G$  is the initial strength of the data points interaction. Although many decreasing functions can be used, Gomez et al only consider  $f(x) = \frac{1}{x^3}$  and  $f(x) = e^{-x^2}$ .

Since RAIN is creating the clusters while it is moving the data points, it is possible to use the number of merged points after some checking iterations for determining if RAIN is using an appropriate value of  $G$ . Algorithm 3 shows the process for determining the initial interaction strength.

Algorithm 4 shows the complete RAIN algorithm. Although it looks like RAIN has linear time complexity, the time complexity of RAIN was estimated as  $O(n\sqrt{n})$  under experimental evidence.

---

**Algorithm 3.** Initial Strength Estimation

---

```

GetInitialStrength(  $x, \Delta(G), M, \varepsilon$ )
1  $G = 1$ 
2  $\text{RGC}(x, G, \Delta(G), M, \varepsilon)$  // test the given strength
3  $K$  =number of merged points
4 while  $\frac{n}{2} - K > \sqrt{N}$  do
5    $G = 2 * G$ 
6    $\text{RGC}(x, G, \Delta(G), M, \varepsilon)$  // test the given strength
7    $K$  =number of merged points
8    $a = \frac{G}{2}$ 
9    $b = G$ 
10   $G = \frac{a+b}{2}$ 
11  $\text{RGC}(x, G, \Delta(G), M, \varepsilon)$  // test the given strength
12  $K$  =number of merged points
13 while  $|\frac{n}{2} - K| > \sqrt{N}$  do
14   if  $\frac{n}{2} > K$  then  $b = G$  else  $a = G$ 
15    $G = \frac{a+b}{2}$ 
16 return  $G$ 

```

---



---

**Algorithm 4.** RAIN Algorithm

---

```

Rain(  $x, \Delta(G), M, \varepsilon, \alpha$ )
1  $G = \text{GETINITIALSTRENGTH}(x, \Delta(G), M, \varepsilon)$ 
2 clusters =  $\text{RGC}(x, G, \Delta(G), M, \varepsilon)$ 
3 return  $\text{GETCLUSTERS}(\text{clusters}, \text{size}(x), \alpha)$ 

```

---

## 4 Incremental Rain: Incremental Approach for Gravitational Clustering

Our incremental clustering technique is based on the divide and conquer approach. The structure of the algorithm can be describe as follows: For a partition of a data set of  $N$  data points, RAIN is executed and a set of clusters is established. Then, for every identified cluster RAIN is executed again to establish sub-clusters. Next, for every sub-cluster a prototype is generated representing the points that belong to the sub-cluster; then a set of representative elements (prototypes) are generated for each cluster. The idea is to maintain the properties of the original cluster by including the data distribution and its effect in the space. In this way, the position of a prototype is the averaged position of the points in the sub-cluster the prototype is represented while the prototype mass is the summation of the masses of the points in the sub-cluster. Similar to the approach presented in [7] and [8], where every point of the data set has an unitary mass at the beginning of the algorithm, prototypes will have the mass of the points that it is representing and new data points will have a unit mass. Clearly, prototypes will exert a greater gravitational force in the space than new points and therefore will have a greater effect over the neighboring points. The Incremental RAIN algorithm (INCRRAIN) is presented in Algorithm 5.

**Algorithm 5.** INCRAIN Algorithm

---

```

INCRAIN(newData,  $\Delta(G)$ ,  $\varepsilon, \alpha$ )
1. prototypes = loadPrototypes()
2. newData = loadNewData()
3. data = newData  $\cup$  { prototypes }
4. prototypes =  $\ddot{i}i_{\frac{1}{2}}$ 
5. clusters1 = RAIN(data,  $\Delta(G)$ ,  $\sqrt{\text{size}(\text{data})}, \varepsilon, 0$ )
6. for j=1 to numClusters1 do{
7.   if clusters1[ j ].getMass()  $\geq \alpha * \text{data.size}()$  then{
8.     //  $\alpha$  : Data Mass percentage
9.     clusters2 = RAIN( clusters[ j ],  $\Delta(G)$ ,  $\varepsilon, \alpha * \text{clusters1}[ j ].\text{getMass}()$ )
10.    for k=1 to numClusters2 do{
11.      prototype = clusters2[ k ].getPrototype()
12.      prototypes = prototypes  $\cup$  { prototype }
13.    }
14.  }
15. }
16. savePrototypes()

```

---

An heuristic was used to determine the number of RAIN iterations to be used with each partition (iteration). According to our experiments, having a partition with  $n$  points, the  $\sqrt{n}$  value proves to be sufficient (line 5) to maintain an acceptable performance, in terms of clustering quality and time complexity, for the algorithm. For every cluster generated after the Rain iteration (line 5), a set of sub-clusters is generated, by applying RAIN again (line 9). These sub-clusters lay the foundations for the construction of the prototypes (line 11). Finally, every prototype is added to the data set returned by the algorithm, these prototypes are labeled and along with a new partition of the data set can be used as input elements for the algorithm (line 16). The way RAIN selects the moving interacting points was changed to handle the concept of mass in the algorithm. As suspected, a point with a greater mass should exert a greater influence in the space than a point with a unitary mass. To represent this influence, a roulette system was implemented. It works by assigning probability of selection to a point depending on its mass. This mechanism allows the prototypes to have a greater impact during the execution of the algorithm. Since the prototype's probability of staying in the system an unlimited amount of time is high, there is a mass decay factor for the prototypes during every iteration. This is done to achieve two important goals: (1) the prototypes are not allowed to grow indefinitely, therefore, avoiding the union of clusters or even worst a "Big Crunch"; (2) the adaptation possibility is open for the algorithm, and it can adjust its learning process as the data patterns change. All of this is possible by diminishing the mass of the prototype. If it is not clustered with any new data point during the execution of the algorithm, the prototype tends to disappear.

## 5 Experiments

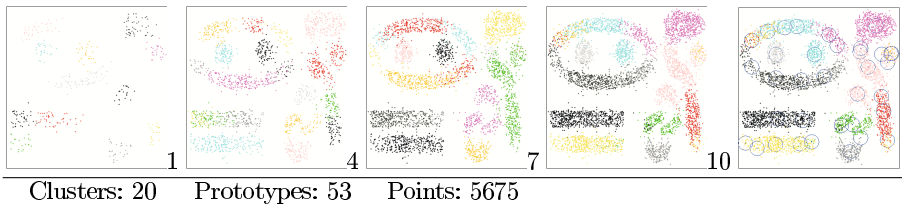
Several experiments were developed using the Chameleon synthetic data set included in the CLUTO toolkit. The RAIN algorithm was executed with the following parameters: Merging distance  $\epsilon = 10e-4$ , Gravitational Constant decay rate  $(\Delta G) = 10e-3$ , Minimum cluster size  $\alpha = 50$  points, RAIN iterations =  $(\sqrt{N})$ , with  $N$  as the number of points in the iteration (partition + prototypes), and the number of RAIN iterations over every cluster to find sub-clusters: =  $(\sqrt{C})/2 + 1$ , with  $C$  as the number of points that belong to the cluster.

Figures 1-2 show the progression of the algorithm with partition sizes of 10% and 20% respectively. For every experiment we show the number of final clusters detected, the number of generated prototypes and the number of points these prototypes represent. To establish the impact of the prototypes in the space we represent the influence of the gravitational force generated by the prototypes by a circular area presented in the figures with blue color. This force was calculated using the equation:

$$F = \frac{m_1 * m_2 * G}{d^3}$$

Here,  $m_1$  is the mass of the prototype,  $m_2$  is the mass of a single point (equal to 1),  $G$  is the gravitational constant used during the execution of the algorithm, and  $d$  is the distance between the prototype and the point. We establish a threshold for the force of 0.5 to establish a standard way to represent the forces. Therefore, the circle represents an area where the force is greater or equal to 0.5.

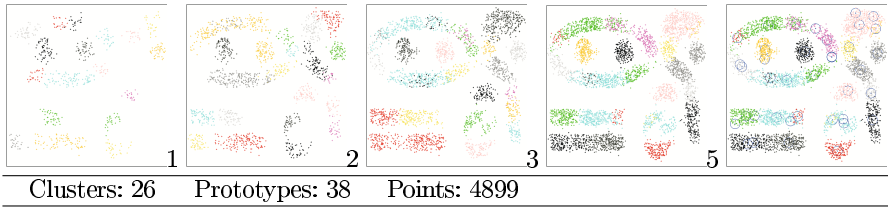
As can be noticed, INCRRAIN is able to detect the cluster by keeping different prototypes for each cluster regardless the amount of data points it is given each time. When new points are given, it is able to combine the previous learned clusters structure with the new data points, see two adjacent images in Figure 1. For example, clusters and prototypes in Figure 1.4 and Figure 1.7 are almost the same but representing more points. Clearly, there is a correspondence between the set of sub-clusters detected in previous iterations with the set of sub-cluster generated in following iterations (a new partition of data points is given). Such



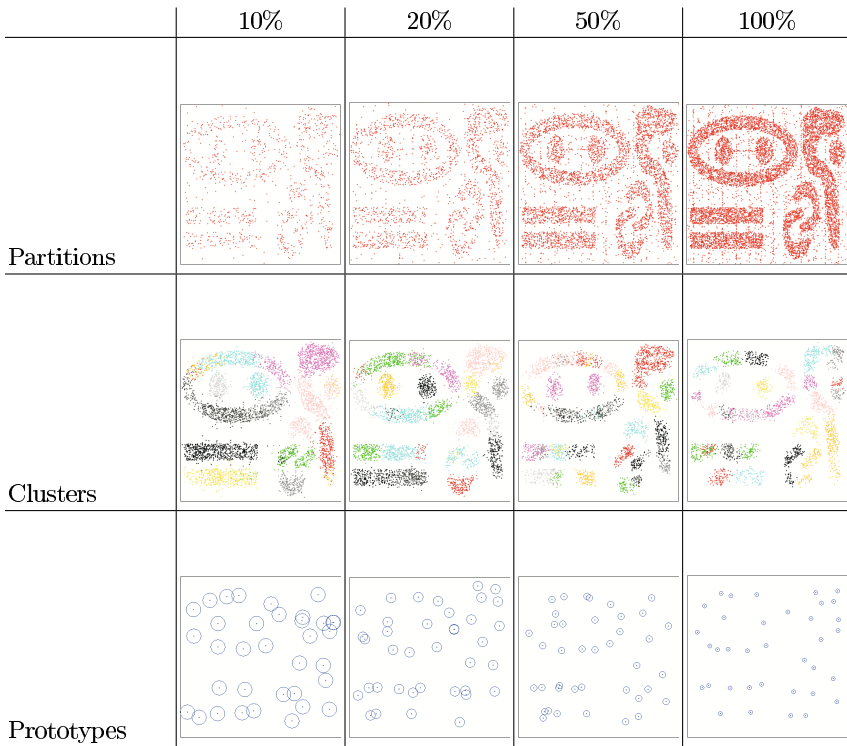
**Fig. 1.** Algorithm Progression - 10% Data Set Partition. Clusters found after showing the first, fourth, seventh, and last partition.

<sup>1</sup> The size of the partition indicates the percentage of the data set that was given to INCRRAIN each time. Therefore, a partition size of 10% indicates that INCRRAIN was run 10 times each time taking a new 10% of the data set.





**Fig. 2.** Algorithm Progression - 20% Data Set Partitions. Clusters found after showing the first, second, third, and last partition.



**Fig. 3.** Partitions, Clusters and Prototypes for diverse partitions sizes

correspondence is also visible when using a 20% of the data set partition and between different partition sizes.

Figures 3 and 4 show the results obtained when the partition size was set to 10%, 20%, 50% and 100% for three different Chameleon data sets. Clearly, the proposed approach is able to detect many of the the clusters but the amount of prototypes and their sizes is very sensible to the size of the partition and the data properties. In the case which the partition data does not allow the clear

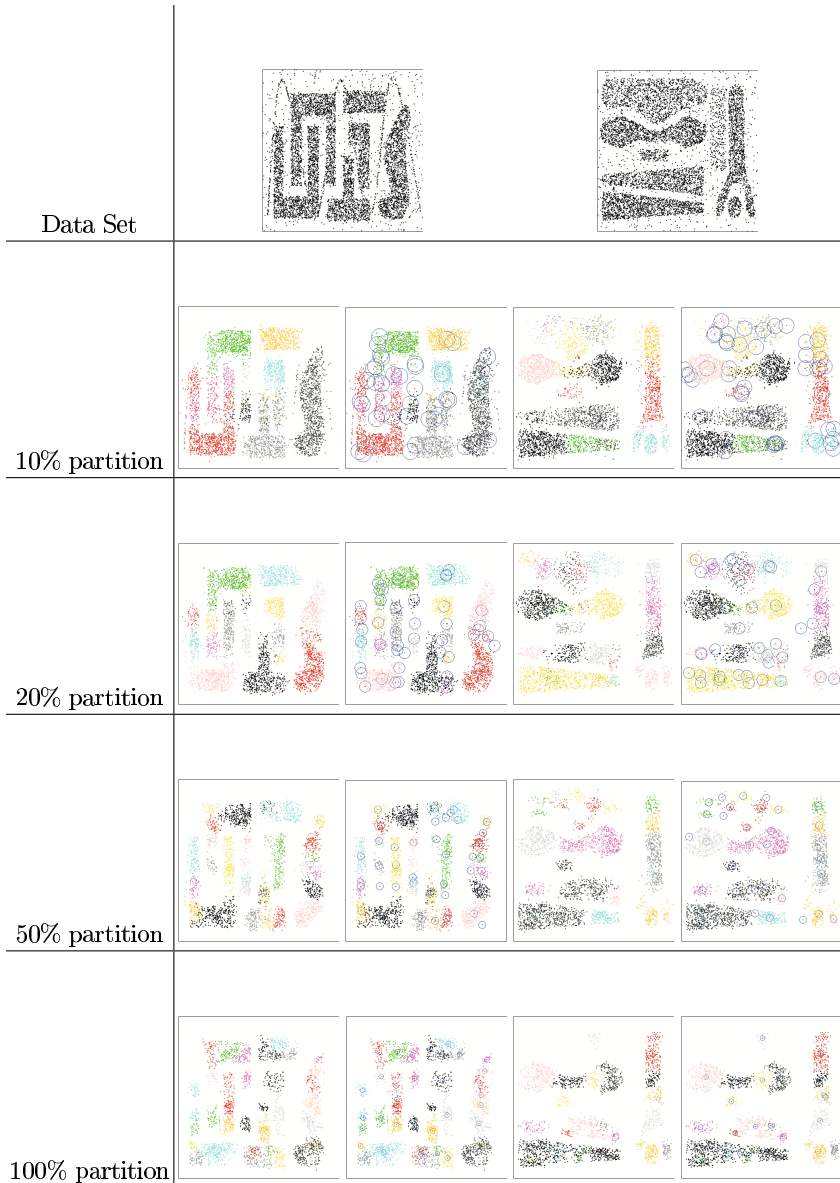


Fig. 4. Experimental Results using different Data Sets

identification of clusters, its is possible that the generated prototypes will not be very reliable or represent the clusters in an unappropriated way. As expected, if the partition size is increased, the set of prototypes gives a better representation of the clusters and the clustering process expends more time.

## 6 Conclusions and Future Work

An incremental technique for the RAIN algorithm was proposed. According to the obtained results, the performance of the proposed approach is acceptable and can be used when the information is gathered from a dynamic environment or when there are not enough resources for managing the full data set. However, there are many factors that can be studied to improve the algorithm behavior such as a better concept for prototypes and prototype sizes (area of influence). Our future work will concentrate in the study of the effect of the number of RAIN iterations over the data set, to obtain a more appropriated set of clusters related to the size of the partition since that parameter seems to be a very determining factor.

## References

1. Antoniol, G., Penta, M.D., Neteler, M.: Moving to smaller libraries via clustering and genetic algorithms (2003)
2. Cormer, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. McGraw-Hill, New York (1990)
3. Dhillon, I., Kogan, J., Nicholas, C.: Feature selection and document clustering. In: Berry, M. (ed.) *A Comprehensive Survey of Text Mining*, Springer, Heidelberg (2003)
4. Ester, M., Kriegel, H., Sander, J., Wimmer, M., Xu, X.: Incremental clustering for mining in a data warehousing environment. In: *Proceedings of 24rd international conference on very large data Bases*, Morgan Kaufmann, San Francisco (1998)
5. Fisher, D.: Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence Research* 4, 147–180 (1996)
6. Fotakis, D.: Incremental algorithms for facility location and k-median. *Theoretical Computer Sciences* 361(2-3), 275–313 (2006)
7. Gomez, J., Dasgupta, D., Nasraoui, O.: A new gravitational clustering algorithm. In: *Proceedings of the Third SIAM International Conference on Data Mining 2003* (2003)
8. Gomez, J., Nasraoui, O., Leon, E.: Rain: Data clustering using randomized interactions of data points. In: *ICMLA 2005, 2005th edn. Proceedings of the Third International Conference on Machine Learning and Applications* (2004)
9. Han, J., Kamber, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco (2000)
10. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. *ACM Computing Surveys* 31(3) (1999)
11. Roure, J., Talavera, L.: Robust incremental clustering with bad instance orderings: A new strategy. In: Coelho, H. (ed.) *IBERAMIA 1998. LNCS (LNAI)*, vol. 1484, pp. 136–147. Springer, Heidelberg (1998)
12. Wang, W., Yang, J., Muntz, R.R.: STING: A statistical information grid approach to spatial data mining. In: Jarke, M., Carey, M.J., Dittrich, K.R., Lochovsky, F.H., Loucopoulos, P., Jeusfeld, M.A. (eds.) *Twenty-Third International Conference on Very Large Data Bases*, Athens, Greece, pp. 186–195. Morgan Kaufmann, San Francisco (1997)
13. Widyantoro, D., Ioerger, T., Yen, J.: An incremental approach to building a cluster hierarchy. In: *Proceedings of the 2nd IEEE International Conference on Data Mining*, pp. 705–708. IEEE Computer Society Press, Los Alamitos (2002)

# On the Influence of Class Information in the Two-Stage Clustering of a Human Brain Tumour Dataset

Raúl Cruz-Barbosa<sup>1,2</sup> and Alfredo Vellido<sup>1</sup>

<sup>1</sup> Universitat Politècnica de Catalunya, Jordi Girona, 08034, Barcelona, Spain  
{rcruz,avellido}@lsi.upc.edu

<sup>2</sup> Universidad Tecnológica de la Mixteca, Car. Acatlima km. 2.5, 69000, Huajuapán, Oaxaca, México

**Abstract.** This paper analyzes, through clustering and visualization, Magnetic Resonance Spectra corresponding to a complex human brain tumour dataset. Clustering is performed as a two-stage process, in which the first stage model is Generative Topographic Mapping (GTM). In semi-supervised settings, class information can be added to refine the clustering process. A class information-enriched variant of GTM, class-GTM, is used here for a first cluster description of the data. The number of clusters used by GTM is usually large for visualization purposes and does not necessarily correspond to the overall class structure. Consequently, in a second stage, clusters are agglomerated using the K-means algorithm with different initialization strategies, some of them defined ad hoc for the GTM models. We aim to evaluate how and under what circumstances the use of class information influences tumour cluster-wise class separability in the final result of the two-stage clustering process.

## 1 Introduction

Medical decision making is usually riddled with uncertainty, especially in sensitive settings such as non-invasive brain tumour diagnosis. Uncertainty may result from lack of information, for instance from unmeasured effects such as metabolic indicators that are not directly accessible with current modalities, or it may arise from inherent variability in the measurements that are available. The latter is of critical importance, as intra- and inter-patient variability within particular diseases are key factors in determining the best discrimination that can be achieved directly from the data.

The data analysed in this study correspond to Magnetic Resonance Spectra (MRS). This technique is of clinical interest especially in cases that remain ambiguous after clinical investigation. Information derived from the MRS can contribute to the evidence base available about the pathology of interest for a particular patient, thus providing specialist support to the clinician [1,2].

The fields of Machine Learning and Statistics coexist with data analysis as a common target. An example can be found in Finite Mixture Models, which have

established themselves as a flexible and robust method for multivariate data clustering [3]. In many practical data analysis scenarios, as medical decision making, these models can be assisted by multivariate data visualization. Finite Mixture Models can be endowed with data visualization capabilities, provided certain constraints are enforced. One alternative is constraining the mixture components to be centred in a low-dimensional manifold embedded into the usually high-dimensional observed data space, as in Generative Topographic Mapping (GTM) [4]. This is a manifold learning model that can be seen as a probabilistic alternative to Self-Organizing Maps (SOM) [5] for simultaneous data clustering and visualization. Class information can be integrated as part of the GTM training to enrich the cluster structure definition provided by the model [6]. The resulting class-GTM model is the basis of this paper.

SOM and GTM do not place any strong restriction on the number of mixture components (or clusters), in order to achieve an appropriate visualization of the data. This richly detailed cluster structure does not necessarily match the more global cluster and class structure of the data. In this scenario, a two-stage clustering procedure may be useful to uncover the global structure of the MRS data [7]. Class-GTM can be used in the first stage to generate a detailed cluster partition in the form of a mixture of components. The centres of these components can be further clustered in the second stage. For that role, the well-known K-means algorithm is used in this study. The first goal of this paper is assessing to what extent the introduction of class information improves the final clusterwise class separation capabilities of the clustering model. The issue remains of how we should initialize K-means in the second clustering stage. Random initialization, with the subsequent choice of the best solution, was the method of choice in [7]. This approach, though, does not make use of the prior knowledge generated in the first stage of the procedure and requires a somehow exhaustive search of the initialization space. Here, we propose two different ways of introducing such prior knowledge in the initialization of the second stage K-means, without compromising the final clusterwise class separation capabilities of the model. This fixed initialization procedures, which result from class-GTM properties, allow significant computational savings.

The rest of the paper is organized as follows: In section 2, we summarily introduce the GTM and its class-GTM variant, as well as the two-stage clustering procedure with its alternative initialization strategies. Several experimental results are provided and discussed in section 3, while a final section outlines some conclusions and directions for future research.

## 2 Two-Stage Clustering

The two-stage clustering procedure outlined in the introduction is described in this section. The first stage model, namely class-GTM, is introduced first. This is followed by the details of different initialization strategies for the second stage. We propose two novel second stage fixed initialization strategies that take advantage of the prior knowledge obtained in the first stage.

### 2.1 The Class-GTM Model

The standard GTM is a non-linear latent variable model defined as a mapping from a low dimensional latent space onto the multivariate data space. The mapping is carried through by a set of basis functions generating a constrained mixture density distribution. It is defined as a generalized linear regression model:

$$\mathbf{y} = \phi(\mathbf{u})\mathbf{W} , \tag{1}$$

where  $\phi$  is a set of  $M$  basis functions  $\phi(\mathbf{u}) = (\phi_1(\mathbf{u}), \dots, \phi_M(\mathbf{u}))$ . For continuous data of dimension  $D$ , spherically symmetric Gaussians of the form

$$\phi_m(\mathbf{u}) = \exp \left\{ -1/2\sigma^2 \|\mathbf{u} - \mu_m\|^2 \right\} \tag{2}$$

are an obvious choice of basis function, with centres  $\mu_m$  and common width  $\sigma$ ;  $\mathbf{W}$  is a matrix of adaptive weights  $w_{md}$  that defines the mapping, and  $\mathbf{u}$  is a point in latent space. To avoid computational intractability a regular grid of  $K$  points  $\mathbf{u}_k$  can be sampled from the latent space. Each of them, which can be considered as the representative of a data cluster, has a fixed prior probability  $p(\mathbf{u}_k) = 1/K$  and is mapped, using (1), into a low dimensional manifold non-linearly embedded in the data space. This latent space grid is similar in design and purpose to that of the visualization space of the SOM. A probability distribution for the multivariate data  $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$  can then be defined, leading to the following expression for the log-likelihood:

$$L = \sum_{n=1}^N \ln \left\{ \frac{1}{K} \sum_{k=1}^K \left( \frac{\beta}{2\pi} \right)^{D/2} \exp \left\{ \frac{-\beta \|\mathbf{y}_k - \mathbf{x}_n\|^2}{2} \right\} \right\} , \tag{3}$$

where  $\mathbf{y}_k$ , usually known as *reference* or *prototype* vectors, are obtained for each  $\mathbf{u}_k$  using (1); and  $\beta$  is the inverse of the noise variance, which accounts for the fact that data points might not strictly lie on the low dimensional embedded manifold generated by the GTM. The EM algorithm is an straightforward alternative to obtain the Maximum Likelihood (ML) estimates of the adaptive parameters of the model, namely  $\mathbf{W}$  and  $\beta$ .

The class-GTM model is an extension of GTM and therefore inherits most of its properties. The main goal of this extension is to improve class separability in the clustering results of GTM. For this purpose, we assume that the clustering model accounted for the available class information. This can be achieved by modelling the joint density  $p(t, \mathbf{x})$ , where  $t$  is a class variable, instead of  $p(\mathbf{x})$ , for a given set of classes  $\{T_i\}$ . For the Gaussian version of the GTM model [6,8], such approach entails the calculation of the posterior probability of a cluster representative  $\mathbf{u}_k$  given the data point  $\mathbf{x}_n$  and its corresponding class label  $t_n$ , or class-conditional *responsibility*  $\hat{z}_{kn}^t = p(\mathbf{u}_k | \mathbf{x}_n, t_n)$ , as part of the E step of the EM algorithm. It can be calculated as:

$$\hat{z}_{kn}^t = \frac{p(\mathbf{x}_n, t_n | \mathbf{u}_k)}{\sum_{k'=1}^K p(\mathbf{x}_n, t_n | \mathbf{u}_{k'})} = \frac{p(\mathbf{x}_n | \mathbf{u}_k) p(t_n | \mathbf{u}_k)}{\sum_{k'=1}^K p(\mathbf{x}_n | \mathbf{u}_{k'}) p(t_n | \mathbf{u}_{k'})} = \frac{p(\mathbf{x}_n | \mathbf{u}_k) p(\mathbf{u}_k | t_n)}{\sum_{k'=1}^K p(\mathbf{x}_n | \mathbf{u}_{k'}) p(\mathbf{u}_{k'} | t_n)} , \tag{4}$$

and, being  $T_i$  each class,

$$p(\mathbf{u}_k|T_i) = \left( \frac{\sum_{n;t_n=T_i} p(\mathbf{x}_n|\mathbf{u}_k)}{\sum_n p(\mathbf{x}_n|\mathbf{u}_k)} \right) \left( \frac{\sum_{k'} \sum_{n;t_n=T_i} p(\mathbf{x}_n|\mathbf{u}_{k'})}{\sum_n p(\mathbf{x}_n|\mathbf{u}_{k'})} \right)^{-1} . \quad (5)$$

Equation (4) differs from the standard responsibility  $\hat{z}_{kn}$  of GTM in that, instead of imposing a fixed prior  $p(\mathbf{u}_k) = 1/K$  on latent space, we consider a class-conditional prior  $p(\mathbf{u}_k|T_i)$ . Once the class-conditional responsibility is calculated, the rest of the model parameters are estimated following the standard EM procedure.

## 2.2 Two-Stage Clustering Based on GTM

In the first stage of the proposed two-stage clustering procedure, a class-GTM is trained to obtain the representative prototypes (detailed clustering) of the observed dataset  $\mathbf{X}$ . As mentioned in the introduction, the number of prototype vectors is usually chosen to be large for visualization purposes, and does not necessarily reflect the global cluster and class structure of the data. In this study, the resulting prototypes  $\mathbf{y}_k$  of the class-GTM are further clustered using the well-known K-means algorithm (a description of which can be found, for instance in [7]). In a two-stage procedure similar to the one described in [7], based on SOM, the second stage K-means initialization in this study is first randomly replicated 100 times, subsequently choosing the best available result, which is the one that minimizes the error function

$$E = \sum_{c=1}^C \sum_{\mathbf{x} \in G_c} \|\mathbf{x} - \mu_c\|^2 , \quad (6)$$

where  $C$  is the final number of clusters in the second stage and  $\mu_c$  is the centre of the K-means cluster  $G_c$ . This approach seems somehow wasteful, though, as the use of GTM instead of SOM can provide us with richer a priori information to be used for fixing the K-means initialization in the second stage.

Two novel fixed initialization strategies that take advantage of the prior knowledge obtained by class-GTM in the first stage are proposed. They are based on two features of the model, namely: the Magnification Factors (MF) and the Cumulative Responsibility (CR). The Magnification Factors measure the level of stretching that the mapping undergoes from the latent to the data spaces. Areas of low data concentration correspond to high distortions of the mapping (i.e., high MF), whereas areas of high data density correspond to low MF. The MF is described in terms of the derivatives of the basis functions  $\phi_j(\mathbf{u})$  in the form:

$$MF = \frac{dA'}{dA} = \det^{1/2} (\psi^T \mathbf{W}^T \mathbf{W} \psi) , \quad (7)$$

where  $\psi$  has elements  $\psi_{ji} = \partial \phi_j / \partial u^i$  [9]. If we choose  $C$  to be the final number of clusters for K-means in the second stage, the first proposed fixed initialization strategy will consist on the selection of the class-GTM prototypes corresponding

to the  $C$  non-contiguous latent points with lowest MF for K-means initialization. That way, the second stage algorithm is meant to start from the areas of highest data density.

The CR is the sum of responsibilities over all points in  $\mathbf{X}$  for each cluster  $k$ :

$$CR_k = \sum_{n=1}^N \hat{z}_{kn}^t . \quad (8)$$

The second proposed fixed initialization strategy, based on CR, is similar in spirit to that based on MF. Again, if we choose  $C$  to be the final number of clusters for K-means in the second stage, the fixed initialization strategy will now consist on the selection of the class-GTM prototypes corresponding to the  $C$  non-contiguous latent points with highest CR. That is, the second stage algorithm is meant to start from those cluster prototypes that are found in the first stage to be most responsible for the generation of the observed data.

## 3 Experiments

### 3.1 Human Brain Tumour Data

MRS is a non-invasive tool capable of providing a detailed fingerprint of the biochemistry of living tissue. The data used in this study consist of 304 single voxel PROBE (PROton Brain Exam system) spectra acquired in vivo for fourteen viable tumour types: meningiomas (58 cases), glioblastomas (86), metastases (38), astrocytomas of 2 (22) and 3 (7) grades, PNETs (9), oligoastrocytomas (6), oligodendrogliomas (7), rare (19), pilocytic astrocytoma (3), malignant lymphomas (10), haemangioblastomas (5), abscesses (8), and schwannomas –4 cases (typology that will be used in this study as class information) and from adjacent normal brain tissue (22). A description of the automated protocol used for the acquisition of these data can be found in [10]. The clinically relevant regions of the spectra were sampled to obtain 200 frequency intensity values. The complexity of the problem, in terms of high dimensionality, was compounded by the small number of spectra available, which is commonplace in MRS data analysis [1]. This makes either clustering or visualization almost compulsory for automated data analysis.

### 3.2 Experimental Design and Settings

The GTM and class-GTM models were implemented in MATLAB®. For the experiments reported next, the adaptive matrix  $\mathbf{W}$  was initialized, following a procedure described in [4], as to minimize the difference between the prototype vectors  $y_k$  and the vectors that would be generated in data space by a partial PCA,  $m_k = V_2 u_k$ , where the columns of matrix  $V_2$  are the two principal eigenvectors (given that the latent space considered here is 2-dimensional). Correspondingly, the inverse variance  $\beta$  was initialised to be the inverse of the  $3^{rd}$



PCA eigenvalue. This ensures the replicability of the results. The value of parameter  $\sigma$ , describing the common width of the basis functions, was set to 1. The grid of latent points  $u_k$  was fixed to a square 12x12 layout. The corresponding grid of basis functions  $\phi$  was equally fixed to a 5x5 square layout.

The goals of these experiments are fourfold. First, we aim to assess whether the inclusion of class information using class-GTM in the first stage of the two-stage procedure results in any improvement in terms of clusterwise class separability (and under what circumstances) compared to the procedure using standard GTM. Second, we aim to assess whether the two-stage procedure improves, in the same terms, on the use of direct clustering of the data using K-means. Third, we aim to test whether the second stage initialization procedures based on MF and CR of the class-GTM, described in section 2.2, retain the class separability capabilities of the two-stage clustering procedure in which K-means is randomly initialized. If this is the case, a fixed second stage initialization strategy should entail a substantial reduction of computational time compared to a random second stage initialization requiring a large number (100 in the reported experiments and also in [7]) of algorithm runs. In fourth place, we aim to explore the properties of the structure of the dataset concerning atypical data. For that, we use a variant of the GTM (with and without class information) that behaves robustly in the presence of outliers: the  $t$ -GTM [11].

The MRS data, described in section 3.1, will be first clustered using both GTM and class-GTM to illustrate the differences between these models. The results will be first compared visually, which should help to illustrate the visualization capabilities of the models. Beyond the visual exploration that could be provided by class-GTM and GTM, the second stage clustering results should be explicitly quantified in terms of clusterwise class separability. For that purpose, the following entropy-like measure is proposed:

$$\begin{aligned}
 E_{G_c}(\{T_i\}) &= - \sum_{\{G_c\}} P(G_c) \sum_{\{T_i\}} P(T_i|G_c) \ln P(T_i|G_c) \\
 &= - \sum_{c=1}^C \frac{K_{G_c}}{K} \sum_{i=1}^{|\{T_i\}|} p_{ci} \ln p_{ci} \quad .
 \end{aligned}
 \tag{9}$$

Sums are performed over the set of classes (tumour types)  $\{T_i\}$  and the K-means clusters  $\{G_c\}$ ;  $K$  is the total number of prototypes;  $K_{G_c}$  is the number of prototypes assigned to the  $c^{th}$  cluster;  $p_{ci} = \frac{K_{G_{ci}}}{K_{G_c}}$ , where  $K_{G_{ci}}$  is the number of prototypes from class  $i$  assigned to cluster  $c$ ; and, finally,  $|\{T_i\}|$  is the cardinality of the set of classes. The minimum possible entropy value is 0, which corresponds to the case of no clusters being assigned prototypes corresponding to more than one class.

Given that the use of a second stage in the clustering procedure is intended to provide final clusters that best reflect the overall structure of the data, the problem remains of what is the most adequate number of clusters. This is a time-honoured matter of debate, which goes beyond the scope of this paper. In this paper we do not use any cluster validity index and we simply evaluate the entropy measure for solutions from 2 up to 15 clusters.

### 3.3 Results and Discussion

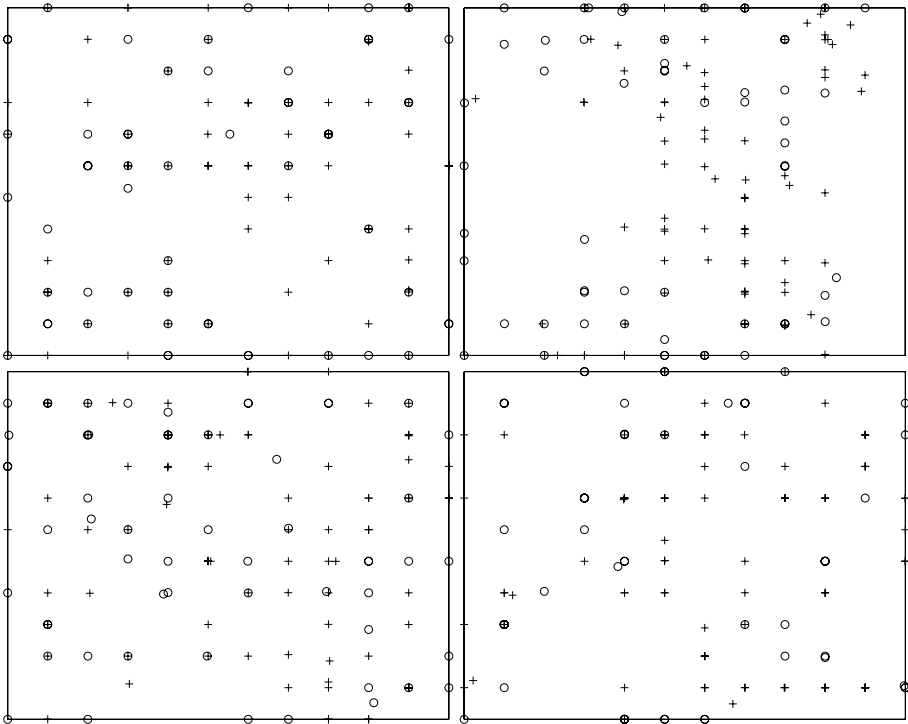
In the first stage of the two-stage clustering procedure, GTM,  $t$ -GTM and their class-enriched variants class-GTM and class- $t$ -GTM were trained to model the human brain tumour dataset described in section 3.1. The resulting prototypes  $\mathbf{y}_k$  were then clustered in the second stage using the K-means algorithm. This last stage was performed in three different ways, as described in section 2.2. In the first one, K-means was randomly initialized 100 times, selecting the results corresponding to the minimum of the error function (6). In the second, we used the Magnification Factors of class-GTM as prior knowledge for the initialization of K-means. In the third, Cumulative Responsibility was used as prior knowledge. In all cases, K-means was forced to yield a given number of final clusters, from 2 up to 15. The final entropy was calculated for all the above procedures and numbers of clusters.

Before considering the entropy results, visualization maps (obtained using the mean of the posterior distribution:  $\sum_{k=1}^K \mathbf{u}_k \hat{z}_{kn}$  or  $\sum_{k=1}^K \mathbf{u}_k \hat{z}_{kn}^t$ ) of all the trained models in the first stage were generated. Three hypotheses are made for the clustering results visualized here. First, the use of class information in the clustering models should yield visualization maps where the classes are separated better than in those models which do not use it. Second, the use of  $t$ -GTM should help to diminish the influence of outliers. Consequently, the visualization maps generated with these models should show the data more homogeneously distributed throughout the visualization maps than in Gaussian GTM models which do not use it. Thirdly, since the tumour dataset is mainly compound of poorly represented classes, we hypothesize that these “small” classes will consist mainly of atypical data.

The clustering model proposed to test the second and third hypothesis is  $t$ -GTM, a variant of the standard GTM that replaces the mixture of Gaussians by a mixture of Student’s  $t$ -distributions, which are known to be best at dealing with atypical data, given their heavier tails. Details on the formulation of  $t$ -GTM can be found in [11] and are omitted here for the sake of brevity. The two-stage clustering experiments were repeated for  $t$ -GTM without class information and for class- $t$ -GTM [6], the corresponding variant of the model with class information.

Given the complexity of the dataset and the space limitation, we only provide one of these illustrative visualizations in Fig. 1. Here, two tumour types (meningioma and glioblastoma, the most represented classes) are shown. The right column of Fig. 1, where the models that include class information are located, suggests that the first hypothesis is sustained, since the class separability between both classes (‘o’ and ‘+’) is better than that of the models that do not make use of class information, located in the left column. This is the result of a more pronounced overlapping of both classes, clearly seen in the left hand-side models of Fig. 1.

The use of  $t$ -distributions in the models represented in the bottom row is more spread throughout the map than that of the Gaussian models of the top row. This is an indication that the  $t$ -GTM models are moderating the effect of



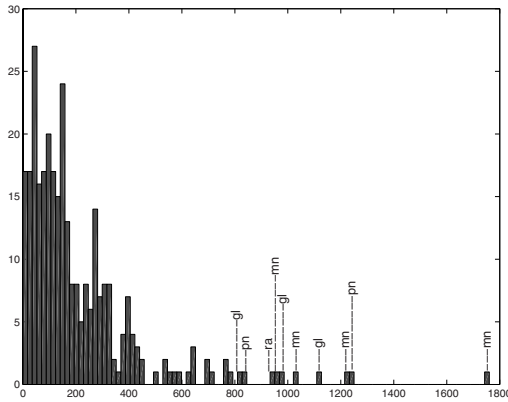
**Fig. 1.** Representation, on the 2-dimensional latent space of GTM and its variants, of part of the tumour data set described in the main text. The representation is based on the mean posterior distributions for the data points belonging to meningioma ('o') and glioblastoma ('+') tumour types. The axes of the plot are the elements of the latent vector  $\mathbf{u}$  and convey no meaning by themselves. For that reason, axes are kept unlabeled. (Top left): GTM without class information. (Top right): class-GTM. (Bottom left):  $t$ -GTM without class information. (Bottom right): class- $t$ -GTM.

outliers. The differences are not huge and, again, this is an indication that there might be not too many outliers in the dataset. All the previous results were generally supported for the rest of the data as well. The two first hypotheses are, therefore, preliminarily supported.

We now turn our attention to the third hypothesis. It was shown in [12] that a given data instance could be characterized as an outlier if the value of

$$O_n^* = \sum_k \hat{z}_{kn} \beta \| \mathbf{y}_k - \mathbf{x}_n \|^2 \tag{10}$$

was sufficiently large. The histogram in Fig. 2 displays the values of  $O_n^*$  from (10) for the brain tumour dataset. First of all, and supporting our previous impression, not too many data could be clearly characterized as outliers according to this histogram. We did the same for the class- $t$ -GTM model and, for illustration,



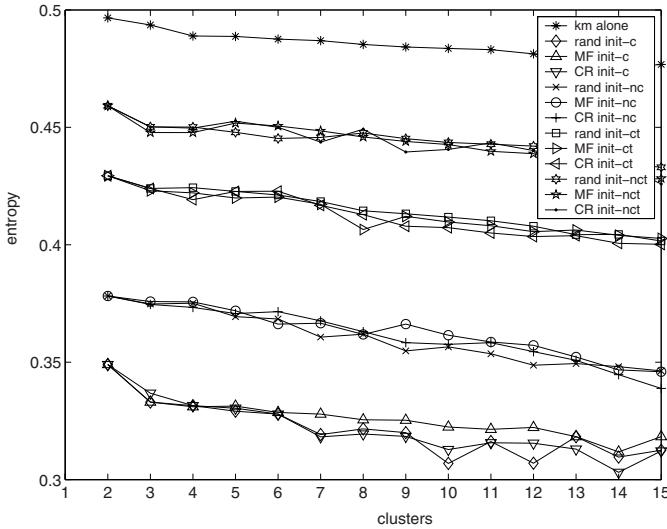
**Fig. 2.** Histogram of the statistic (10); outliers are characterized by its large values. For illustration, the ten largest values are labeled. See tumour type acronyms in Table 1

**Table 1.** Outlier count and percentage (in brackets) by tumour type (see figures in section 3.1) given a threshold for (10)

Tumour type	# of outliers (%): <i>t</i> -GTM	# of outliers (%): class- <i>t</i> -GTM
Meningioma ( <i>mn</i> )	6 (10.3%)	4 (6.9%)
Glioblastoma ( <i>gl</i> )	6 (7.0%)	5 (5.8%)
Metastases ( <i>me</i> )	1 (2.6%)	2 (5.3%)
Astrocytoma 2 ( <i>a2</i> )	1 (4.5%)	4 (18.2%)
PNET ( <i>pn</i> )	2 (22.2%)	2 (22.2%)
Rare ( <i>ra</i> )	2 (10.5%)	2 (10.5%)
Lymphoma ( <i>ly</i> )	1 (10.0%)	0 (0.0%)
Haemangioblastoma ( <i>hb</i> )	1 (20.0%)	1 (20.0%)

proposed an artificial threshold. The 20 largest values of  $O_n^*$  were taken as outliers. The results are summarized in Table 1. Surprisingly, given the complex tumour typology of the dataset, these results do not support the third hypothesis, as many outliers belong to the tumour types with better representation in the dataset (*mn*, *gl*, *me*, and *a2*).

The entropy measurements quantifying the clusterwise class separation for the brain tumour dataset are shown in Fig. 3. Two immediate conclusions can be drawn. Firstly, all the two-stage clustering procedures based on class-GTM perform much better than the direct clustering of the data through K-means, in terms of class separation, but also better than the two-stage procedure without class information based on the standard GTM. Secondly, random initialization in the second stage of the clustering procedure, with or without class information, does not entail any significant advantage over the proposed fixed initialization strategies across the whole range of possible final number of clusters, while being far more costly in computational terms.



**Fig. 3.** Entropies for the clustering of the tumour dataset using two-stage clustering with different initializations (based on MF (MF init), CR (CR init) and random (rand init)), and K-means alone. The ‘c’ symbol means that the corresponding model using class information was used in the first stage and ‘nc’ for the opposite. The ‘t’ in the legend label means that *t*-GTM was used in the first stage.

The entropy measure in (9) quantifies the level of agreement between the clustering solutions and the class distributions. In terms of the overall class separation provided by the clustering models, it has been shown that the addition of class information consistently helps.

### 4 Conclusion

In this paper we have carried out an analysis of the influence exerted by the inclusion of class information in the two-stage clustering of a complex human brain tumour dataset. We have also tested different strategies of initialization for the second stage of this procedure. The first stage is based on the manifold learning class-GTM model, which, besides clustering, also provides visualization of the data and clusters on a low-dimensional space. The second stage is based on the well-known K-means algorithm, which was initialized either multiple times randomly or, making use of the prior knowledge provided by class-GTM in the first stage, in a fixed manner using a novel procedure based on its Magnification Factors and Cumulative Responsibility. The reported experiments have shown that the two-stage random and fixed initializations yield almost identical results in terms of clusterwise class separation, with the former being computationally more costly. It has also been shown that the two-stage clustering procedures based on standard GTM and class-GTM perform better than the direct K-means

clustering of the data in terms of this clusterwise class separation and that the inclusion of class information improves the clusterwise class separation. The latter would help in semi-supervised settings, in which tumours without type label coexist with the labelled ones. The existence of atypical data or outliers in the human brain tumours MRS dataset under study, and its influence on the clustering process, have also been explored.

**Acknowledgements.** Alfredo Vellido is a researcher within the Ramón y Cajal program of the Spanish MEC and acknowledges funding from the MEC I+D project TIN2006-08114. Raúl Cruz-Barbosa acknowledges SEP-SESIC (PROMEP program) of México for his PhD grant.

## References

1. Lisboa, P.J.G., Vellido, A., Wong, H.: Outstanding issues for clinical decision support with Neural Networks. In: Malmgren, H., Borga, M., Niklasson, L. (eds.) *Artificial Neural Networks in Medicine and Biology*, 2000, pp. 63–71. Springer, Heidelberg (2000)
2. Lisboa, P.J.G., El-Deredy, W., Lee, Y.Y.B., Huang, Y., Corona-Hernandez, A.R., Harris, P.: Characterisation of brain tissue from MR spectra for tumour discrimination. In: Yan, H. (ed.) *Signal Processing for Magnetic Resonance Imaging and Spectroscopy*, Marcel Dekker, New York, pp. 569–588 (2002)
3. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(3), 381–396 (2002)
4. Bishop, C.M., Svensén, M., Williams, C.K.I.: The Generative Topographic Mapping. *Neural Computation* 10(1), 215–234 (1998)
5. Kohonen, T.: *Self-Organizing Maps*. Springer, Heidelberg (1995)
6. Cruz, R., Vellido, A.: On the improvement of brain tumour data clustering using class information. In: *STAIRS 2006. Proceedings of the 3rd European Starting AI Researcher Symposium*, Riva del Garda, Italy (2006)
7. Vesanto, J., Alhoniemi, E.: Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks* (2000)
8. Sun, Y., Tiño, P., Nabney, I.T.: Visualization of incomplete data using class information constraints. In: Winkler, J., Niranjana, M. (eds.) *Uncertainty in Geometric Computations*, pp. 165–174. Kluwer Academic Publishers, Dordrecht (2002)
9. Bishop, C.M., Svensén, M., Williams, C.K.I.: Magnification Factors for the GTM algorithm. In: *Proceedings of the IEE fifth International Conference on Artificial Neural Networks*, pp. 64–69 (1997)
10. Tate, A.R., Majós, C., Moreno, A., Howe, F.A., Griffiths, J.R., Arús, C.: Automated classification of short echo time in In Vivo  $^1\text{H}$  brain tumor spectra: a multicenter study. *Magnetic Resonance in Medicine* 49, 29–36 (2003)
11. Vellido, A.: Missing data imputation through GTM as a mixture of  $t$ -distributions. *Neural Networks* 19(10), 1624–1635 (2006)
12. Vellido, A., Lisboa, P.J.G.: Handling outliers in brain tumour MRS data analysis through robust topographic mapping. *Computers in Biology and Medicine* 36(10), 1049–1063 (2006)

# Learning Collaboration Links in a Collaborative Fuzzy Clustering Environment

Rafael Falcon<sup>1</sup>, Gwanggil Jeon<sup>2</sup>, Rafael Bello<sup>1</sup>, and Jechang Jeong<sup>2</sup>

<sup>1</sup> Computer Science Department, Universidad Central de Las Villas  
Carretera Camajuaní km 5 ½ Santa Clara, Cuba  
{rfalcon, rbello}@uclv.edu.cu

<sup>2</sup> Dept. of Electronics and Computer Engineering, Hanyang University  
17 Haengdang-dong, Seongdong-gu, Seoul, Korea  
windcap315@ece.hanyang.ac.kr

**Abstract.** Revealing the common underlying structure of data spread across multiple data sites by applying clustering techniques is the aim of collaborative clustering, a recent and innovative idea brought up on the basis of exchanging information granules instead of data patterns. The strength of the collaboration between each pair of data repositories is determined by a user-driven parameter, both in vertical and horizontal collaborative fuzzy clustering. In this study, Particle Swarm Optimization and Rough Set Theory are used for setting the most suitable values of the collaboration links between the data sites. Encouraging empirical results uncovered the deep impact observed at the individual clusters, allowing us to conclude that the overall effect of the collaboration has been improved.

**Keywords:** collaborative fuzzy clustering, information granules, collaboration links, particle swarm optimization, rough set theory, evolutionary computation.

## 1 Introduction

Uncovering natural associations of data patterns is known as data clustering [1]. A milestone in this field was achieved with the introduction of fuzzy clustering [2], allowing the representation of many real-life situations where patterns are to be classified in more than one subset. Although the prominence of clustering in data analysis and visualization has become evident for a long time, in recent years we have witnessed a rise in the emergence of innovative approaches, most of them related to fuzzy clustering. Kernel-based fuzzy clustering [3] and self-organized fuzzy clustering [4] are two illustrative and supportive examples of this claim.

One of the most appealing approaches related to fuzzy clustering is concerned with the extension of clustering to several data sites. This methodology was named “collaborative clustering” and was originally researched by Pedrycz [5]. The goal of an overall discovery of the common underlying structure in an ensemble of data sites led to the need of exchanging some type of information between the data repositories in order to have it exercise an influence over the formation of clusters at each

individual location. The sharing restrictions imposed on patterns on account of many possible reasons (security, privacy, etc.) were gracefully overcome by exchanging information granules instead, namely membership degrees of patterns to clusters. One step forward in this direction came with the development of a hybrid rough-fuzzy collaborative architecture [6] wherein a sound, seamless blending of both fuzzy and rough sets gave rise to another way of carrying out the collaboration, this time exchanging and even moving cluster prototypes from one site to another.

Back to its inception, collaborative fuzzy clustering straightly depends on setting some user-driven parameters which may have a greater or lesser impact on the ultimate outcome of the algorithm. This is the case of the distance function used to compare data patterns, the desired number of clusters ( $c$ ) for partitioning the data or the termination criterion. But none of these is most critical than the values ascribed to the collaboration matrix; that is, settling in advance the strength of the collaboration for each pair of datasets, which obviously might spoil the whole collaborative scheme by driving it to an unfruitful state, since two datasets keeping a poor or none resemblance in terms of their clustering results might be commanded to collaborate to a high extent or, on the other hand, the collaboration may almost be nullified for a couple of datasets having a great similarity in their clustering outcomes.

Things get worse as more data repositories are engaged in the process. Some hints have been issued, such as the admission of a certain value (say  $p\%$  of the change in membership of the partition matrices prior and after the collaboration). However, no reliable and verifiable alternative appears in literature today. Due to the relevance of this fact, we would like to contribute with providing an overall sense of completeness to such a brilliant idea as the collaborative fuzzy clustering is.

In a few words, applying rough set theory to this framework allows to define the lower and upper approximations for a given data site. The collaboration is then restricted to those sites which are similar to one another in terms of their clusters. This saves an extra, non-trivial task of broadcasting partition matrices from a repository to another, which in noisy, low-bandwidth scenarios is indeed profitable. After the application of rough sets, initial values for the collaboration links are set and subsequently refined by means of Particle Swarm Optimization (PSO). The impact of these modifications to the early collaborative fuzzy clustering environment is clearly remarkable through experiments carried out with both real and synthetic data.

The study is structured as follows: a brief explanation of the main concepts and notations of both horizontal and vertical collaborative clustering is the subject of the next section. Rough sets and Particle Swarm Optimization are briefly reviewed in Sections 3 and 4, respectively. The outline of our proposal for computing the collaboration links is thoroughly exhibited in Section 5 whereas Section 6 deals with the empirical demonstrations asserting the feasibility of the suggested ideas over the collaborative system. Concluding remarks are depicted in Section 7.

## 2 Collaborative Fuzzy Clustering: Concepts and Notations

As previously explained, collaborative fuzzy clustering was introduced in order to spread the clustering scheme to multiple repositories holding data of the same nature. A blueprint of the overall collaborative system is pictured in Figure 1.



Basically, we have  $P$  subsets of data from which we would like to obtain  $c$  fuzzy clusters. For such purpose, a fuzzy clustering algorithm must be selected (the standard FCM [7] is a good candidate) as well as a distance function. This algorithm is executed individually on each data site, yielding a list of prototypes  $v[ii]$  and a partition matrix  $U[ii]$ , where  $ii$  is the  $ii$ -th dataset. The termination criterion is the one imposed by the clustering method, generally the failure in minimizing its objective function for two consecutive iterations.

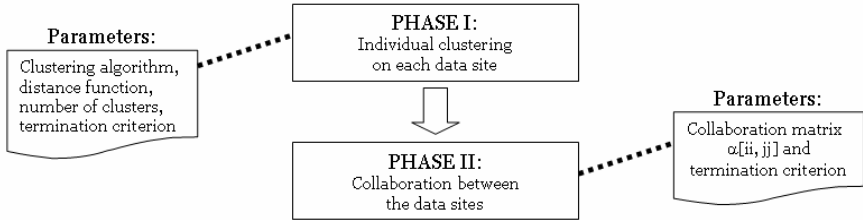


Fig. 1. The two phases of the collaborative clustering and their associated parameters (degrees of freedom)

Once the first stage is over, we set up the values of the collaboration links (entries of the matrix  $\alpha[ii, jj]$ ) and start passing back and forth the partition matrices between the data sites. Thereby, the confidentiality constraint over the data patterns remains enforced and the membership degrees of patterns to clusters act as information granules.

Two major collaboration modes have been identified: horizontal and vertical. This section is devoted to explain the main features of the horizontal mode, while the earnest reader can delve into the details of the vertical mode by going over [8].

### 2.1 Horizontal Collaborative Fuzzy Clustering

This collaboration mode relies on the same group of  $N$  patterns across all data sites which are split in disjoint subsets of features [9], as it is portrayed in Figure 2. Therefore, a single pattern can be retrieved by concatenating the corresponding subpatterns.

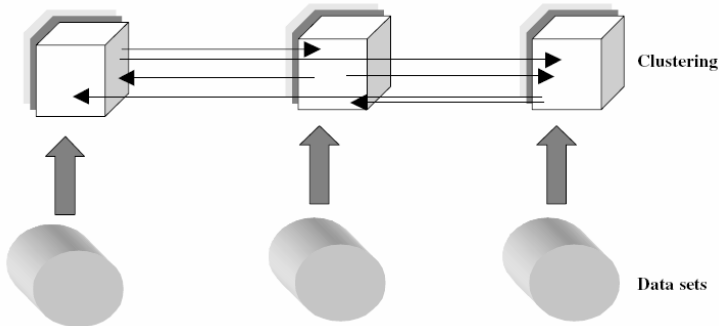


Fig. 2. A general illustration of the workflow of horizontal clustering

Let's say you own data on different countries which are described by subsets of economical, social, political and alike indicators held in different data locations. We are interested in flagging the country with an overall score of 'Good', 'Fair', 'Bad'. This can be achieved by running the clustering algorithm on each data site with  $c = 3$  and later on sharing the findings of each data location with one another.

Let us denote the dimensionality of the patterns (the number of features) in each data subset as  $n[ii]$ , where  $ii$  is the  $ii$ -th data site,  $ii = 1..P$  and the distance function between the  $i$ -th cluster prototype and the  $k$ -th pattern in the same dataset as  $d_{ik}[ii]$ ,  $i = 1..c$  and  $k = 1..N$ . Additionally, the membership degree of the  $k$ -th pattern to the  $i$ -th cluster in the  $ii$ -th dataset is represented by  $u_{ik}[ii]$ .

Being this so, the objective function to be minimized is displayed in (1):

$$Q [ii] = \sum_{k=1}^N \sum_{i=1}^c u_{ik}^2 [ii] d_{ik}^2 [ii] + \sum_{\substack{jj=1 \\ jj \neq ii}}^P \alpha [ii, jj] \sum_{k=1}^N \sum_{i=1}^c (u_{ik} [ii] - u_{ik} [jj])^2 d_{ik}^2 [ii]. \tag{1}$$

The role of the second term in the above expression is to have the  $ii$ -th dataset become fully cognizant of what's going on in the remaining subsets. The optimization details are omitted for the sake of brevity. For a thorough and insightful explanation of what collaborative clustering looks like and how it behaves, check [8].

### 2.2 Evaluation of the Collaborative Phenomenon of Clustering

Assessing the real impact the collaboration exercises over the data sets is possible via two major perspectives: the level of data (comparing the cluster prototypes as well as their composition by patterns) and the level of information granules (that is, partition matrices exchanged between the databases).

In his seminal paper [5], Pedrycz introduced two evaluation functions which are complementary and have a strong dependence on the collaboration matrix chosen for the computations. They are defined in (2) and (3).

$$\delta = \frac{1}{N P c} \sum_{ii=1}^P \sum_{jj=i+1}^P \sum_{k=1}^N \sum_{i=1}^c | u_{ik} [ii] - u_{ik} [jj] | \tag{2}$$

$$\Delta = \frac{1}{N P c} \sum_{ii=1}^P \sum_{k=1}^N \sum_{i=1}^c | u_{ik} [ii] - u_{ik} [ii - ref] | \tag{3}$$

Some notations must be made clear: let  $P$  be the number of data sites,  $N$  the number of patterns,  $c$  the number of clusters,  $ii$  represents the  $ii$ -th data site after the collaboration,  $ii-ref$  is the partition matrix of the  $ii$ -th site prior to the collaboration and  $u_{ik}$  is the membership degree of the  $k$ -th pattern to the  $i$ -th cluster.

While a strong collaboration significantly lowers the value of the  $\delta$  function (as can be remarked from the definition), it does the opposite with the value of  $\Delta$ , for the

semantic meaning of this function is the overall departure of the referential structure with no collaboration.

### 3 Rough Set Theory

Rough set theory is a mathematical approach to vague and uncertain knowledge originally introduced by Pawlak [10]. The rough set methodology is based on the assumption that there exists a certain amount of information associated to every object of the universe which is described by means of some attributes.

The crux of this approach is to approximate a given concept (a set of objects  $X$  of some universe  $U$ ) by means of two crisp sets named lower and upper approximations, respectively. This is to be done after defining an indiscernibility relation  $R \subseteq U \times U$  over a set of attributes  $B$ . If  $R$  is an equivalence relation, it induces a partition of the universe into blocks of indiscernible objects that can be used to build knowledge on a real or abstract world. Later on, the equivalence classes are computed for each object of the universe.

$$B(x) = \{y \in U: yRx\} . \tag{4}$$

After that, the lower and upper approximations are calculated as read below:

$$B_*(X) = \{x \in U: B(x) \in X\} . \tag{5}$$

$$B^*(X) = \{x \in U: B(x) \cap X \neq \phi\} . \tag{6}$$

The lower approximation contains all objects which can be classified with full certainty as members of the given concept  $X$  whereas the upper approximation stores those objects that are possible members of  $X$ . The set resulting of the difference between the upper and lower approximations is called the “boundary region” and is denoted as  $BND(X)$ . It holds those objects which cannot be categorized for sure as members of the concept. This is the way rough set theory quantifies vagueness. A concept  $X$  having an empty boundary region is said to be crisp (exact) with respect to the subset of attributes  $B$ ; otherwise, it is said to be rough (inexact) with regards to  $B$ .

### 4 Particle Swarm Optimization

Within the evolutionary computational methods, Particle Swarm Optimization [11] [12] is considered as a landmark for optimizing continuous functions. It emerged as an agent-based approach simulating the social, cognitive behavior of bird flocks and fish schools. Here, each solution is represented by a particle. Particles group in “swarms” (one or more, depending on the implementation) and the swarm’s evolution to the optimal solutions is supported by each particle’s velocity and position update equations (7) and (8).

$$v_{ik} = wv_{ik} + c_1 U(0,1) (pbest_{i,k} - x_{i,k}) + c_2 U(0,1) (gbest_k - x_{i,k}) \tag{7}$$

$$x_{i,k} = x_{i,k} + v_{i,k} \tag{8}$$

The first term in (7) is affected by an inertia weight ( $w$ ) which greatly determines the local versus global exploration rate of the particle, the second one represents the cognitive component the particle has on its best position reached so far. Finally, the third term models the social behavior of the swarm, wherein each particle attempts to come close to the best particle's (leader's) position. Further details about the PSO algorithm can be found anywhere in literature. Lots of modifications to the original procedure have been proposed in order to speed up the convergence rate towards the optimal solution or to adjust the algorithm to have it cope with a great deal of problems[13] [14].

### 5 Our Approach: Setting the Collaboration Links

The problem of seeking the most suitable values of the entries of the collaboration matrix  $\alpha$  is undertaken as a two-stage approach. The first one deals with an initial determination of the collaboration links by using a rather straightforward rough set methodology. After that, such values are submitted to a PSO-driven tuning process whose output is the collaboration matrix intended to optimize a certain fitness function. The two original phases of the collaborative system envisioned by Pedrycz seamlessly mix with and complement the steps of our approach, as displayed in Figure 3.

The first stage undergoes no alterations and is carried out once a fuzzy clustering approach (Fuzzy C-Means is a good candidate) and a distance function are picked out.

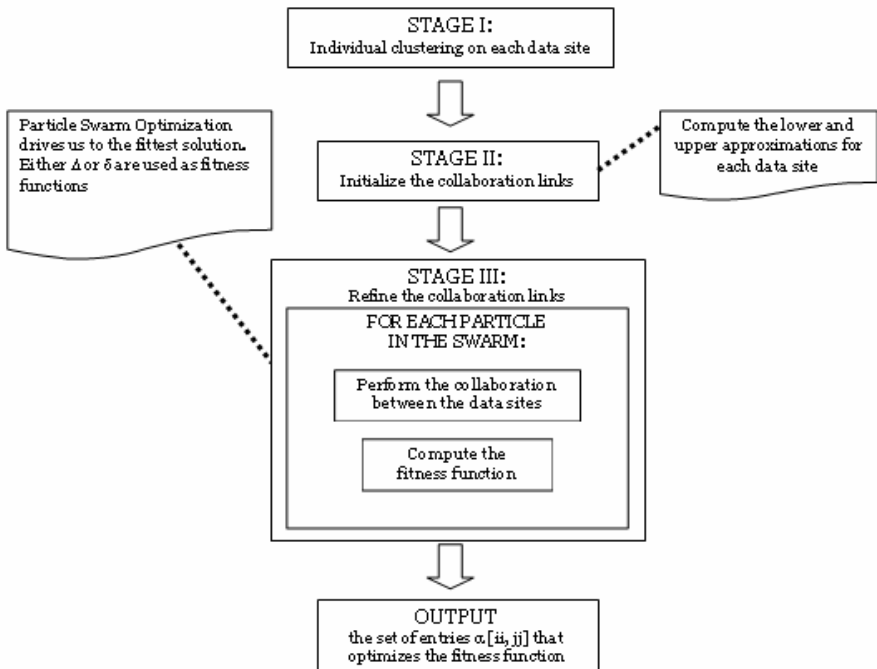


Fig. 3. The new collaborative scheme for learning the collaboration matrix

### 5.1 Initializing the Collaboration Links

The main principle behind the scene here is the assumption that the collaboration needs to be restricted to the sites that are worth cooperating with. That is, it is pointless to exchange information granules with a data set which is a priori known to have a quite different nature than ours. On the other hand, two very similar datasets will not yield a fruitful outcome after collaborating either, so they are commanded not to do it. In both cases, it translates into less network traffic since fewer partition matrices are flooding the network.

The underlying principles of the rough set theory perfectly fit in order to allow the collaboration to occur between some data sites alone. Everything starts by defining the following indiscernibility relation  $R$ :

$$R(A, B) = \frac{1}{c} \sum_{i=1}^c \frac{\max_{j=1 \dots c} \{ |C_i[A] \cap C_j[B]| \}}{|C_i[A]|} . \tag{9}$$

Where  $C_i[A]$  represents the  $i$ -th cluster in site  $A$  and  $|x|$  is the cardinality (number of elements) of set  $x$ .

The above relation intersects every cluster of site  $A$  with each cluster of site  $B$  and adds up the maximum cardinality found for each cluster in  $A$ . This can be interpreted as a similarity degree between two subsets of data by looking at the patterns forming each cluster in each data set. A comparison between the cluster prototypes in each site is not feasible in the horizontal mode of collaborative clustering as every subset of data has its own different feature space. However, this is the right way to proceed in the vertical mode.

Let us now introduce a “rough threshold”  $\rho \in [0, 1/2]$  for defining the approximations of a data site (10) and (11) as follows:

$$R_*(A) = \{B \in U_p: R(A, B) \geq 1 - \rho\} . \tag{10}$$

$$R^*(A) = \{B \in U_p: R(A, B) \geq \rho\} . \tag{11}$$

Where  $U_p$  is the set of all data sites,  $|U_p| = P$ . Notice that the property  $R_*(A) \subseteq R^*(A)$  holds and the boundary region of  $A$  can be defined as  $BND(A) = \{B \in U_p: \rho < R(A, B) < 1 - \rho\}$ . Hence, we limit the collaboration only to the sites belonging to the boundary region of  $A$ , as shown below:

$$\alpha[A, B] = \begin{cases} R(A, B) & \text{if } B \in BND(A) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Yet the question on how to choose a suitable value of  $\rho$  remains open and this is left to the user. Such value can be interpreted as the restriction level for the collaboration between the repositories to happen. The greater the value of  $\rho$ , the more restricted the collaboration is. We could generalize this standpoint by allowing two thresholds  $0 < \theta < \tau < 1$  to denote the admissible boundaries of the degrees of similarity between two datasets for the collaboration to be carried out.

## 5.2 Tuning the Collaboration Links

Expression (12) sets the collaboration links within the interval  $[\rho, 1 - \rho] \subseteq [0, 1]$ . Nevertheless, a good collaborative effect is frequently attained by taking the  $\alpha$  values out of  $[0, 1]$ . This is precisely the purpose of using the PSO algorithm for the tuning stage.

In this configuration, each particle (prospective solution) represents a collaboration matrix  $\alpha$  for our problem. The chosen implementation is the ‘global best’ approach in which all particles belong to a single swarm.

Taking advantage of the previously initialized collaboration matrix, the initial position of one particle in the swarm is assigned to it, which surely will speed up the convergence rate of the PSO algorithm and reduce the number of iterations needed to find the optimal solution. The rest of the particles get random initializations for their positions and velocities with 0 as the lower bound and not having an upper bound.

This is in regards to the initialization of the particles’ positions and velocities. Once the PSO algorithm gets the ball rolling, no constraint is imposed neither over the positions nor over the velocities of the particles.

In each iteration, all particles are to perform the second phase of the original collaborative scheme (that is, the collaboration between the data sites) so as to compute the prototypes and partition matrices on the basis of the collaborative links each particle represents. A schematic representation of this process is depicted in figure 3.

As previously stated, the fitness function for the PSO algorithm can be either  $\Delta$  or  $\delta$ . In each case, however, the optimization and its associated meaning are different.

After a number of predefined iterations the PSO algorithm stops, returning the best particle found so far as the optimal solution. The experimental section further elaborates on the configuration of the PSO procedure.

## 5.3 The PSORS-CFC Algorithm Outlined

The entire aforementioned scheme is properly outlined as an algorithm in a pseudo-language for later implementation purposes:

```
Algorithm PSORS_CFC;
Given: P data sites holding subsets of patterns
Select: number of clusters (c), distance function,
clustering algorithm (CA), rough threshold  $\rho$ , number of
particles N, inertia weight W, acceleration constants
c1 and c2, termination criterion.
```

1. For each data site:
  - a) Apply the CA algorithm to perform an individual clustering of the data into c clusters;
  - b) Reckon the similarity degree  $R(A, B)$  with respect to all other datasets as defined in (9);
  - c) Compute the lower and upper approximations using (10) and (11);

2. Initialize the collaboration matrix  $\alpha$  by (12);
3. Use the matrix in step 2) to initialize a single particle in the swarm. Randomly initialize the remaining  $N-1$  particles in the interval  $[0, \infty]$ .
4. Repeat for each particle in the swarm
  - a) Perform the collaboration between the data sites.
  - b) Compute the fitness function (either  $\delta$  or  $\Delta$ ).
  - c) Adjust each particle's position and velocities via the PSO procedure.

Until the termination criterion is satisfied

Output: The optimal collaboration matrix found.

The termination criterion relies on either a predefined maximum number of iterations or an admissible value of the fitness function to be reached.

## 6 Experimental Studies

The experiments were carried out using both synthetic data and publicly available data from the UCI Machine Learning repository. As to the fitness function for the PSO algorithm, both  $\Delta$  and  $\delta$  functions were used indistinctly.

For each selected data set, the whole collaborative scheme was run and the collaboration matrices coming out from the two stages (initialization and tuning) were properly presented in tables.

### 6.1 Synthetic Patterns

A small, synthetic data set consisting of ten patterns was split into  $P = 3$  data sites holding 2-D patterns each. This was thus done in order to visualize the effect the collaboration exercises over the formation of the clusters in each individual repository.

For this sample, the threshold of the initialization phase of the collaboration links was set to  $\rho = 0.15$ , signifying that any pair of data sites whose previous resemblance (in terms of the clusters composition) falls below 0.15 (pointing out that they belong to a very different nature) or above 0.85 (they can be considered similar to a high extent) is commanded not to collaborate, therefore dismissing the anticipated effect of the cooperation and exchange between the sites.

As to the PSO algorithm, the swarm was made up of 10 particles and the iterative procedure stopped after 20 iterations. The inertia weight was taken as 1.4 and dynamically varied as in [15], whereas the acceleration constants  $c_1 = c_2 = 2$ . The achieved results (collaboration links) are depicted in Table 1.

**Table 1.** The collaboration links for the synthetic data computed in each phase of the full collaborative system

Features	$\alpha[ii, jj]$ After Initializing	Fitness	$\alpha[ii, jj]$ After Tuning	Fitness
$P = 3$ $(2 + 2 + 2)$ $\rho = 0.15$ $Patterns = 10$ $c = 3$	$\begin{bmatrix} 0.00 & 0.72 & 0.69 \\ 0.69 & 0.00 & 0.50 \\ 0.67 & 0.50 & 0.00 \end{bmatrix}$	$\Delta = 0.9831371$	$\begin{bmatrix} 0.00 & 64.29 & 97.76 \\ 16.18 & 0.00 & 36.29 \\ 101.21 & 59.36 & 0.00 \end{bmatrix}$	$\Delta = 0.9914646$
		$\delta = 0.0322926$	$\begin{bmatrix} 0.00 & 207.60 & 109.80 \\ 379.84 & 0.00 & 94.32 \\ 369.75 & 178.08 & 0.00 \end{bmatrix}$	$\delta = 0.001034763$

**6.2 Publicly Available Data**

The ‘Boston housing’ and ‘Ionosphere’ databases can be freely downloaded from the UCI Machine Learning repository at <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>. The first one contains 14 features and 506 patterns concerning some characteristics of the housing values in suburbs of Boston. The last feature (price of real state) was isolated in a single data set whereas the rest of the attributes composed the other data file.

Establishing the same configuration of the PSO-refining stage as with the previous experiment (synthetic data), we arrive at the outcomes displayed in Table 2.

**Table 2.** The collaboration links for the ‘Boston housing’ database

Features	$\alpha[ii, jj]$ After Initializing	Fitness	$\alpha[ii, jj]$ After Tuning	Fitness
$P = 2$ $(13 + 1)$ $\rho = 0.15$ $Patterns = 506$ $c = 3$	$\begin{bmatrix} 0.00 & 0.71 \\ 0.71 & 0.00 \end{bmatrix}$	$\Delta = 0.4700127$	$\begin{bmatrix} 0.00 & 28.96 \\ 60.64 & 0.00 \end{bmatrix}$	$\Delta = 0.5333447$
		$\delta = 0.0128494$	$\begin{bmatrix} 0.00 & 181.24 \\ 1622.48 & 0.00 \end{bmatrix}$	$\delta = 0.000130190$

The other selected database was ‘Ionosphere’ which holds 351 patterns, 34 linear attributes and a two-valued class attribute. The data gathered here has to do with complex electromagnetic signals and their associated pulse time and pulse number. Because of the amount of features, we unfolded it into 6 data locations, each one’s patterns located in a feature space with different dimensionality.

The tuning phase was conducted in accordance with the previous configuration used for the aforementioned experiments in contrast to the initialization phase, where the collaboration threshold was raised to  $\rho = 0.25$ . See the results in Tables 3 & 4 below.



**Table 3.** The collaboration links for the ‘Ionosphere’ database (initialization phase)

Features	$\alpha[ii, jj]$ After Initializing	Fitness
$P = 6$ (6+3+5+8+10+3)	$\begin{bmatrix} 0.00 & 0.71 & 0.00 & 0.68 & 0.56 & 0.63 \\ 0.73 & 0.00 & 0.00 & 0.00 & 0.55 & 0.61 \\ 0.00 & 0.00 & 0.00 & 0.74 & 0.60 & 0.63 \\ 0.68 & 0.74 & 0.75 & 0.00 & 0.62 & 0.57 \\ 0.55 & 0.57 & 0.56 & 0.63 & 0.00 & 0.58 \\ 0.60 & 0.63 & 0.62 & 0.59 & 0.56 & 0.00 \end{bmatrix}$	$\Delta = 1.158804$
$\rho = 0.25$		$\delta = 0.4020191$
Patterns = 351		
$c = 3$		

**Table 4.** The collaboration links for the ‘Ionosphere’ database (tuning phase)

Features	$\alpha[ii, jj]$ After Tuning	Fitness
$P = 6$ (6+3+5+8+10+3)	$\begin{bmatrix} 0.00 & 84.84 & 0.00 & 27.05 & 119.13 & 42.67 \\ 121.88 & 0.00 & 0.00 & 0.00 & 84.19 & 52.85 \\ 0.00 & 0.00 & 0.00 & 81.74 & 99.97 & 77.43 \\ 69.42 & 11.07 & 10.20 & 0.00 & 30.50 & 12.06 \\ 85.23 & 50.98 & 53.88 & 56.31 & 0.00 & 23.22 \\ 66.47 & 86.57 & 67.43 & 20.14 & 70.83 & 0.00 \end{bmatrix}$	$\Delta = 1.164955$
$\rho = 0.25$		$\delta = 0.0016727$
Patterns = 351		

### 6.3 Discussion

While the above databases all have a different, distinctive nature when regarding the amount of patterns and features they are composed of as well as the ensemble of subsets into which they were unfolded, an underlying, common behavior is observed throughout the experiments.

First, when computing the lower and upper approximations for each data site during the initialization phase, the chosen threshold  $\rho = 0.15$  for the first two experiments do not allow us to discard any fruitless collaboration apart from the one between each site and itself (trivial, isn't it?). But this is not the case as for the ‘Ionosphere’ database, wherein a slight increase of  $\rho$  to 0.25 prevented us from applying the collaboration scheme in five repositories which were determined to be very similar (without including the six trivial entries  $\alpha[ii, ii]$ ). In any case, the a priori determination of the collaborative links (even when they do not escape from  $[0, 1]$ ) by following a heuristic measure related to the closeness of two sites supposes an

important advantage to the problem. The user may sigh relieved that he is no longer requested to fix such values in advance.

Yet the really interesting phenomenon arises during the tuning stage. Here, the common behavior repeated across the experiments (as you can straightforwardly notice by looking at the tables above) is that the values of  $\Delta$  (which is concerned with the difference between the partition matrices of a certain data set before and after collaboration) are not significantly improved by the PSO algorithm. This drives us to corroborate the empirical observation made in [5] that there exists a level of saturation for the parameter  $\alpha$  beyond which it makes no sense to keep on collaborating with the other data site. Moreover, the PSO approach ran into the group of saturating alphas before arriving to the 10-th iteration (50%). From that point on, the fitness function value was never improved.

The same does not hold when considering the  $\delta$  function (quantifying the collaborative effect by means of the partition matrices of all data sites once the collaboration occurs). The runs of the PSO meta-heuristic confirm that it is possible to continue lowering this value at the expense of increasing the collaboration links (which translates into a thicker blending of the patterns of each data site with those coming from one another). You can check the truthfulness of this claim by looking at the far larger values of  $\alpha$  reported in comparison to the ones displayed when using the  $\Delta$  function. If there exists a saturation level for  $\delta$ , 20 iterations of the PSO algorithm are not enough to find it.

Perhaps the smartest thing one could think of to avoid this effect is considering several fitness functions that represent desirable properties of the collaboration (both at the data and information granules levels) and launch a search of the collaborative links in this scenario.

## 7 Conclusions and Future Work

The impact of the collaboration matrix over the overall effect of the collaboration is clearly distinguishable since the very inception of collaborative fuzzy clustering. In this paper, we have confined ourselves to the horizontal mode of collaborative clustering and outlined a methodology which finds the most suitable set of collaborative links for a predetermined evaluation measure. The proposed approach consists of two phases: the initialization phase (where rough sets are employed) and the tuning phase (governed by the PSO meta-heuristic). The empirical results have been discussed with an open mind, yielding a satisfactory outcome and enticing us to define some data-level comparison functions to measure the collaborative impact coming from other data sets. Subsequently, the extension of the PSO procedure to cope with a multi-objective optimization setting wherein two or more measures are to be optimized at the same time is highly advisable and encouraged.

The idea exhibited here fits the vertical collaborative clustering scheme as well. In this case, we anticipate that the indiscernibility relation used for building the lower and upper approximations can be stated in terms of the distance between the cluster prototypes (since the feature space remains the same for every data set under consideration).

**Acknowledgments.** The support provided by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-005-J04101) is gratefully acknowledged.

## References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. *ACM Computing Surveys* 31(3), 264–323 (1999)
2. Sato-Ilic, M., Jain, L.: Introduction to Fuzzy Clustering. In: Sato-Ilic, M., Jain, L. (eds.) *Innovations in Fuzzy Clustering. Studies in Fuzziness and Soft Computing*, vol. 205, pp. 1–8. Springer, Heidelberg (2006)
3. Sato-Ilic, M., Jain, L.: Kernel based Fuzzy Clustering. In: Sato-Ilic, M., Jain, L. (eds.) *Innovations in Fuzzy Clustering. Studies in Fuzziness and Soft Computing*, vol. 205, pp. 89–104. Springer, Heidelberg (2006)
4. Sato-Ilic, M., Jain, L.: Self Organized Fuzzy Clustering. In: Sato-Ilic, M., Jain, L. (eds.) *Innovations in Fuzzy Clustering. Studies in Fuzziness and Soft Computing*, vol. 205, pp. 125–150. Springer, Heidelberg (2006)
5. Pedrycz, W.: Collaborative Fuzzy Clustering. *Pattern Recognition Letters* 23(14), 1675–1686 (2002)
6. Mitra, S., Banka, H., Pedrycz, W.: Rough-Fuzzy Collaborative Clustering. *Man and Cybernetics* 36(4), 795–805 (2006)
7. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. In: *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press (1981)
8. Pedrycz, W.: *Knowledge-Based Clustering: From Data to Information Granules*. John Wiley and sons, Chichester (2005)
9. Pedrycz, W., Vukovich, G.: Clustering in the Framework of Collaborative Agents. *IEEE International Conference on Fuzzy Systems* 1, 134–138 (2002)
10. Pawlak, Z.: Rough sets. *International Journal of Computer and Information Sciences* 11(1), 341–356 (1982)
11. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. *IEEE International Conference on Neural Networks* 4, 1942–1948 (1995)
12. Kennedy, J., Eberhart, R.: *Swarm Intelligence*. Morgan Kaufmann, San Francisco (2001)
13. Santana-Quintero, L., Ramírez-Santiago, N., Coello, C., Molina, J., García, A.: A New Proposal for Multiobjective Optimization using Particle Swarm Optimization and Rough Set Theory. In: Runarsson, T.P., Beyer, H.-G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) *Parallel Problem Solving from Nature - PPSN IX. LNCS*, vol. 4193, pp. 483–492. Springer, Heidelberg (2006)
14. Omran, M., Engelbrecht, A., Salman, A.: Dynamic Clustering using Particle Swarm Optimization with Application in Unsupervised Image Classification. *Computing and Technology* 9, 199–204 (2005)
15. Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R.: Feature Selection Based on Rough Sets and Particle Swarm Optimization. *Pattern Recognition Letters* 28(4), 459–471 (2006)

# Algorithm for Graphical Bayesian Modeling Based on Multiple Regressions

Ádamo L. de Santana, Carlos Renato L. Francês, and João C. Weyl Costa

Laboratory of High Performance Networks Planning, Federal University of Para  
R. Augusto Correa, 01, 66075-110, Belem, PA, Brazil  
{adamo, rfrances, jweyl}@ufpa.br

**Abstract.** One of the main factors for the knowledge discovery success is related to the comprehensibility of the patterns discovered by applying data mining techniques. Amongst which we can point out the Bayesian networks as one of the most prominent when considering the easiness of knowledge interpretation achieved. Bayesian networks, however, present limitations and disadvantages regarding their use and applicability. This paper presents an extension for the improvement of Bayesian networks, incorporating models of multiple regression for structure learning.

## 1 Introduction

Bayesian networks stand as one of the best computational intelligence techniques among the existing paradigms, being nowadays one of the best methods for treating uncertainty in the field of artificial intelligence [1]. Particularly due to their exceptional analytical properties to represent domains, correlate and study the dependences among its variables, allowing a more easily visualization and understanding of the relations among the variables, consisting on a decisive factor and of great value for the representation and analysis of the domain by the users.

However, just like any other computational algorithm, the Bayesian networks also present limitations and disadvantages, regarding their use as well as their applicability. Among these restrictions, we can point out: the difficulty to correlate variables considering the time factor, or yet, by the absence of models that would allow a deeper the use of its information and results, such as establishing the optimal combination of parameters to achieve a certain requirement. Such studies are of utter importance, given that they constitute on problems and needs observed on real world domains, and which are, eventually, crucial for the decision making process.

This paper presents a new and optimized method for learning the graphical representation of Bayesian networks, in which the creation of the network and the correlation analysis among the variables are made applying multiple regression models.

The paper is organized as follows: in section 2, some related works to the structure learning of Bayesian networks are shown. In section 3, the algorithm for structure learning based on multiple regression is presented. The final remarks of the paper are presented in section 4.

## 2 Previous and Related Works

This section we will present some of the works found in literature and that also served as basis as well as comparison for the studies presented in this paper. The works are also divided here according to the fundamentals of their approaches: whether it is based on the graphical structure learning of the network; on the search for the best configuration, that is, the set of actions or inferences and furthermore its singular values to achieve a specific state; or the temporal analysis for Bayesian networks.

The graphical learning, the construction of a Bayesian network involves the learning of the network structure and the definition of the probabilities associated with its variables. This process can be done directly with the help of experts in the studied domain or automatically, with learning algorithms, which we will focus here. The learning algorithms can be classified as being constraint based, where the structure is obtained by identifying the dependencies among the variables; or through a search and score of the best network structure.

Here, the search and score approach will be used for the learning of the network topology. The search and score works searching through the space of possible existing structures, starting from a graph with no arcs and adding new ones, calculating a score for the given structure until no new arc can be added.

In [2] a search and score method to induce Bayesian networks is proposed, using both fuzzy systems and genetic algorithms. It is proposed a scoring metric based on the evaluation of different quality criteria, which is computed by the fuzzy system; using the genetic algorithm as means to search through the space of possible structures, which has also been applied to the learning of Bayesian networks [3].

The fuzzy system uses as input metrics the Bayesian measure, the minimum description length principle [4], Akaike information criteria [5], and the estimated classification accuracy of the network; thus providing the quality of the network as output. The genetic algorithm is used to search the possible network structures.

Comparatives as to the algorithm performance with well-known algorithms (BayesN [6], Bayes9 [7], Tetrad [8] and K2 [9]), which will also be presented as comparative in section 3, are also shown.

The use of this approach brings however some limitations such as the fact of it being sensitive to the selection of the initial population (for the genetic algorithm) as well as for the different membership functions (for the fuzzy system).

Other recent methods implemented for the learning of the Bayesian graphical structure, usually based on hybrid models can also be seen in [10], [11] and [12], each with its own metric of scoring and evaluation: use of (semantic) crossover and mutation operators to help the evolution process, penalty measure, and Minimum Description Length metric, respectively; [12] proposal however does not involve a need for a complete ordering of the variables as input. Further use of genetic algorithms can also be seen in [13] and [14].

In [15], the use of a previous ordering of the variables is also studied, proposing a multi-phase approach for the graphical learning based on the use of distinct but easy to implement algorithms, which involves a search method for optimal parents to build the structure, followed by a method to eliminate existing cycles in the graph and an finally an evaluation of the network using structural perturbation.

Aside from the ordering of nodes, the dataset (here we will treat only with fully observed cases) size is also an important aspect when considering the network quality and convergence speed of the algorithm. Especially since it is NP-hard [16], exponentially increasing the searching space with the number of variables.

A more thorough overview on the techniques and algorithms for the learning of Bayesian networks can be seen in [17].

### 3 Structure Learning Based on Multiple Regressions

The algorithm presented here searches for the best configuration, amongst the space of possible structures, for the construction of a Bayesian network from the analysis of existing dependences and independences between the variables. The algorithm uses the search and score method, analyzing all the possible graphical combinations that can be set from the variables of the domain. It will be assumed here, at first, the need for an ordinance of the variables; which, though some recent algorithms work without this need, they are not, usually, very efficient [17].

The structure learning is an important problem to be studied, motivated by the fact that the search space of possible structures increases exponentially with the number of variables of the model. This exponential growth can be calculated as follow [18]:

$$G(n) = \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} G(n-i) \tag{1}$$

Equation (2) calculates the number of possible directed acyclic graphs  $G$ , that can be formed with a number of  $n$  variable. Table 1 [19] presents the number of possible graphs (values of  $G$ ) as  $n$  increases.

**Table 1.** Values for  $G$  obtained as  $n$  increases

$n$	$G(n)$
1	$1.0 \times 10^0$
2	$3.0 \times 10^0$
5	$2.9 \times 10^4$
10	$4.2 \times 10^{18}$
20	$2.3 \times 10^{72}$
50	$7.2 \times 10^{424}$
100	$1.1 \times 10^{1631}$

Here, the analysis for the search of the best Bayesian network that represents the domain is made with the use of multiple regressions [20] [21]. The technique of multiple regression denotes a specific model of multivariate analysis.

The models of multivariate analysis are used to adequately study the multiple relations existent among the variables of a domain, in order to obtain a more complete

and realistic understanding in the decision making process [20]. With the use of multiple regressions the changes in the dependent variable can be predicted, in response to the changes in the independent variables.

The search method of the algorithm follows from the ordinance of the variables, where for each attribute  $X_i$  the possible dependencies of the variable with its precedents are examined (variables parents -  $Pa_i$ ), adding arcs between them and verifying the quality of the network created according to its score; continuing, as follows, with the search of another attribute, that added to the previous one(s) would increase the score of the network.

The validation of the network, created by each new added arc, is made through regressions, that can be single (when analyzing the relation with only one variable) or with multiple variables, as it is usually applied.

This algorithm of Multiple Regressions for Structure Learning (MRSL) attributes the score of each network through the value found by the adjusted coefficient of each regression ( $\bar{R}^2$ ); which is obtained as described next.

### 3.1 Modeling and Structure of the Algorithm

Assuming a database  $D$  with  $n$  records and  $i$  number of variables, we are searching for the best Bayesian network structure  $B_S$  for it. We denote the target variable that we are analyzing as  $Y_i$ , the  $k$  variables candidates for parents as  $X_{iA}$ , the  $A_k$  parameters to be estimated and the random errors as  $u_i$ , the generalized formula of the multiple regression model can be specified as follow:

$$Y_i = A_0 + A_1X_{1i} + A_2X_{2i} + \dots + A_kX_{ki} + u_i \tag{2}$$

The general system of the multiple regression can then be seen as a matricial system and represented according (3).

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & X_{12} & \dots & X_{1k} \\ 1 & X_{21} & X_{22} & \dots & X_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & \dots & X_{nk} \end{bmatrix} \times \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_k \end{bmatrix} + \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_k \end{bmatrix} \tag{3}$$

This way,  $Y = XA + u$ , where:  $Y$  is a column vector, with dimension  $n \times 1$ ;  $X$  is a matrix of size  $n \times k$ , that is, with  $n$  observations and  $k$  variables; with the first column representing the intercept  $A_0$ ;  $A$  is a vector with  $k \times 1$  unknown parameters; and  $u$  is a vector with  $n \times 1$  disturbances.

This specification intends to generate the parameters of vector  $A$ ; which can be estimated as follow:

$$A = (X^t X)^{-1} \times X^t Y \tag{4}$$

With the values of  $A$ , the value of the regression coefficient ( $R^2$ ) can then be calculated according to:

$$R^2 = \frac{A^t(X^t X)A - n\bar{Y}^2}{y^t y} \tag{5}$$

Where  $\bar{Y}$  is the mean value of variable  $Y$ , and  $y$  is obtained by the subtraction of  $Y$  by  $\bar{Y}$ . And thus calculating its adjusted value by:

$$\bar{R}^2 = 1 - \left(1 - R^2\right) \left(\frac{n-1}{n-k}\right) \tag{6}$$

In the same way, the absence of dependencies ( $Pa_i = \phi$ ) for the variable in question is assigned when obtained, for each possible relation of dependence, a value close to or below zero for  $\bar{R}^2$ .

Another important aspect is regarding the relevance of the inclusion of new variables in the model. This analysis is made in order to verify whether or not the inclusion of one or more arcs for the variable is indeed relevant for the model, even though with this inclusion a higher  $\bar{R}^2$  might be obtained. The analysis of this aspect is due, in particular, to the fact of being verified during the evaluation tests of the algorithm, that the  $\bar{R}^2$  obtained for the best  $Pa_i$  configuration for the variable  $X_i$  with a number of arcs  $x$  was very close to the one achieved by the best  $Pa_i$  configuration with a number of arcs  $x + 1$ . The same behavior was also observed when comparing the latter with the value obtained with a number of arcs  $x + 2$ , and so on.

In order to provide the analysis relevance to be generally applicable for datasets disregarding their sizes, the  $F$  test, whose formula is presented below, was used to evaluate the contribution of the new variables added to the model.

$$F = \frac{(R_U^2 - R_R^2)/m}{(1 - R_U^2)/(n - k)} \tag{7}$$

Where  $R_U^2$  and  $R_R^2$  are the  $\bar{R}^2$  values obtained for the unrestricted (with the inclusion of the new variables) and restricted (without the inclusion of the variables) regressions, respectively, and  $m$  is the number of variables added to the model. The statistic distribution  $F$  follows with  $m$  and  $n - k$  degrees of freedom. If the  $F$  statistics presents a value different from zero, the added variable is accepted as a possible *parent* variable.

### 3.2 Optimizations Studied

The MRSL algorithm acts in an optimized way, with respect to performance, when compared with other existing learning algorithms in the literature. It works directly without considering the number of states of the variables, not suffering from any combinatorial impact that can be implied by them in the search and score of the best network structure. In order to further optimize the performance of the algorithm some considerations and heuristics can also be adopted. In the very first iterations of each variable, a control can be included in order to decrease the combinatorial space to be covered and, consequentially its execution time.



Firstly, from the values obtained in the correlations of degree one (number of parents equals to one) of the variable  $X_i$  with its precedents  $(X_1, \dots, X_{i-1})$ , it is already possible to observe which, amongst the variables, presents a higher level of correlation with  $X_i$ . This is important as, whenever a new arc can be added in the network structure, the new combination of parents found will have as component, compulsorily, the attribute (or combination of attributes, if the number of arcs is higher than 2) found previously. Thus, only the future regressions for models having as component the nodes assigned previously will be made.

Not only that, but if in the correlations coefficients  $\bar{R}^2$  present values with low significance or close to or below zero, the search for a better configuration and admission of new arcs can cease, as the following ones will also obey the same trend.

Another aspect that can be manipulated, is the indication by the user specialist in the domain of a minimum degree of significance to be verified for the admission of a new arc in the structure.

As described thus far, the algorithm fixates the attributes as the dependent variables and study its relations with the preceding variables according to the ordering set as input, also applying strategies to diminish the combinatory search space. The causality analysis of the algorithm will be further focused now, presenting a means to model the network without the need for a previous ordering of the variables.

### 3.3 Causality

The regression model implemented studies the correlation and dependence among the variables, having the ordering of the variables as main aspect to attribute the direction of the dependence. The sequential order of the variables would be, at this point, very important, given that, by itself, the dependence relations among the variables does not necessarily implies on a causal relation.

The study of causality is then applied with the analysis of Granger [22]. Considering the hypothesis that  $X$  can cause  $Y$  ( $X \rightarrow Y$ ), the test is established between a restricted regression, in which  $Y$  is a function of only its past values; and an unrestricted regression, where  $Y$  is a function of the values of  $Y$  and  $X$ . The functions for the restricted and unrestricted regressions are represented in (8) and (9), respectively.

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \dots + \alpha_s Y_{t-s} + \varepsilon \tag{8}$$

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \dots + \alpha_s Y_{t-s} + \beta_1 X_{t-1} + \dots + \beta_s X_{t-s} + \varepsilon \tag{9}$$

The hypothesis of causality from the analysis of both regressions is then made with the  $F$  test (Equation 7). So that, if the calculated value of  $F$  is higher than its critical values [21] the causality from  $X$  to  $Y$  is accepted.

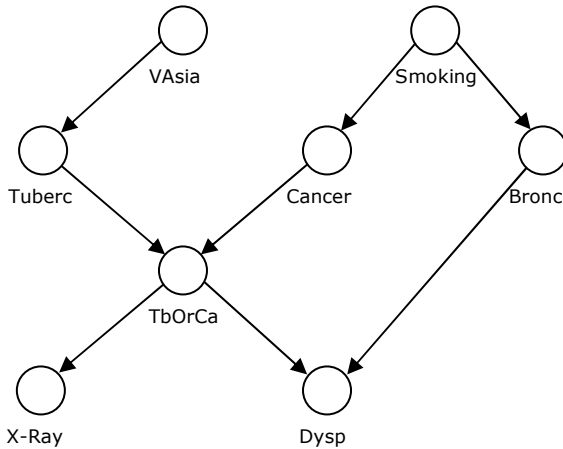
With the advent of the causality analysis, it is possible to establish a new search heuristic, incorporating it with the model detailed on section 3.1, without the need of a previous ordering of the variables. Initially verifying the variables from which the attribute present dependences, and then studying the direction of its causality.

### 3.4 Performance Evaluation

The evaluation of the model was made considering two aspects: the quality of the Bayesian network found by the algorithm, that is, the representativity of the network regarding the domain; and its computational performance.

For comparing the analysis regarding the quality of the generated network, the *Chest Clinic* [23] database was used as application example (usually known as *Asia*), which denotes a problem of a fictitious medical diagnosis, of whether a patient has tuberculosis, lung cancer or bronchitis, based on his X-Ray, dyspnea, visit to Asia and smoking status. The database possesses 8 binary variables and its Bayesian network presents 8 arcs connecting them (Figure 1).

The database was submitted to our learning algorithm to obtain the structure of the Bayesian network. For a quality comparative of the generated network, results from other search and score algorithms were used; the algorithms used were the K2, created by Cooper and Herskovitz (1992), which to this date is still a great reference among the existing algorithms for learning of Bayesian networks, being one of the most trustworthy and successful learning algorithms [24]. The results of the following algorithms existing in literature were also used as comparative for the quality of the generated network for the *Asia* database: Tetrad [8], Bayes9 [7], BayesN [6] and Genetic-Fuzzy [2].



**Fig. 1.** Bayesian network of the Asia database

Table 2 compares the result achieved by our algorithm (MRSL) with the one from the original (the *golden network*, as it is also referenced in literature) Bayesian network of the Asia database, as well as the results obtained by others five existing algorithms in the literature (K2, Tetrad, Bayes9, BayesN and Genetic-Fuzzy). The *Total* column presents the number of arcs found by each algorithm. The column *Correct* contains the number of arcs that were correctly found. The column *Additional*

presents the number of arcs that were found and that are not in the original network. And the column *Absent* presents the number of arcs that were not found and are present in the original network.

**Table 2.** Comparative of the results obtained for the Asia database

Algorithms	Total	Correct	Additional	Absent
<i>MRSL</i>	8	8	0	0
<i>K2</i>	8	7	1	1
<i>Genetic-Fuzzy</i>	9	8	1	0
<i>BayesN</i>	8	5	3	3
<i>Bayes9</i>	4	4	0	4
<i>Tetrad</i>	4	4	0	4

Our algorithm presented results better than the other algorithms, finding an identical structure to the golden network of the *Asia* database, with no additional nor absent arcs. Being followed by the results presented by the *K2* and *Genetic-Fuzzy* algorithms.

Table 3 presents the results for the structure learning of the *Asia* network with datasets of different sizes; now studying the capacity of the algorithm for obtaining the original structure of the network when different volumes of data are available.

**Table 3.** Results for the structures obtained according to the amount of data available

Num. of records	Total	Correct	Additional	Absent
<i>100</i>	9	7	2	1
<i>200</i>	7	7	0	1
<i>500</i>	8	7	1	1
<i>1,000</i>	8	8	0	0
<i>5,000</i>	8	8	0	0
<i>10,000</i>	8	8	0	0

The results obtained (Table 3) showed that the algorithm has a good capacity of learning, even when working with a small amount of data, finding the original structure of the network with a data amount of 1,000 records onwards.

For the performance evaluation of the algorithm, the analysis was made using as testbed experiment the model presented by the *Asia* network, which is composed of 8 variables and 1,000 records, comparing the results obtained with the ones presented by the *K2* algorithm.

The tests made here seeks to verify the performance of the algorithm using as parameter the discretized states of the database variables, that is, the number of possible states that each attribute can assume. The performance tests were made analyzing the execution time for both algorithms over the database, with the attributes (initially binary) discretized from the two initial states until a maximum of ten. The obtained results (Table 4) denote the execution times of the algorithms without considering the time spent for reading the database into the memory.

**Table 4.** Execution times (in seconds) obtained by the algorithms

<b>Num. of states</b>	<b>MRSL</b>	<b>K2</b>
2	0.08	0.1
3	0.08	0.14
4	0.08	0.24
5	0.08	0.51
6	0.08	1.35
7	0.08	3.51
8	0.08	9.10
9	0.08	20.05
10	0.08	44.48

Tables 5 and 6 present the same tests, now also considering an increase in the number of records of the database to 5,000 and 10,000 respectively. Table 7 presents the values, considering only the discretization space of 10, for a better visualization of the gradual behavior in the increase of the execution time between the algorithms.

**Table 5.** Execution times (in seconds) obtained by the algorithms for 5,000 records

<b>Num. of states</b>	<b>MRSL</b>	<b>K2</b>
2	0.42	0.48
3	0.42	0.68
4	0.42	0.93
5	0.42	1.32
6	0.42	2.26
7	0.42	4.53
8	0.42	10.27
9	0.42	21.32
10	0.42	45.42

**Table 6.** Execution times (in seconds) obtained by the algorithms for 10,000 records

<b>Num. of states</b>	<b>MRSL</b>	<b>K2</b>
2	0.84	0.96
3	0.84	1.33
4	0.84	1.78
5	0.84	2.32
6	0.84	3.39
7	0.84	5.81
8	0.84	11.74
9	0.84	22.87
10	0.84	47.05

**Table 7.** Execution times (in seconds) obtained with a number of states set to 10

<b>Num. of records</b>	<b>MRSL</b>	<b>K2</b>
1000	0.08	44.48
5000	0.42	45.42
10.000	0.84	47.05

As it could be verified by the obtained results, the structure learning algorithm based on multiple regressions outperforms on both aspects analyzed: with respect to the quality of the Bayesian network induced by the algorithm as well as in its computational performance. The algorithm uses in its structure statistical models with a fundamental theory, especially concerning the prediction and correlation analysis of the variables; and that, due to its nature, works in an optimized way, improving the performance as the number of states assumed for the variables increases; which is a common characteristic for databases that represent real world domains.

## 4 Final Remarks

The possibility to represent graphically the structure of the patterns obtained from the data, as well as the exploratory character of the analysis allowed by the Bayesian networks, enables to indicate more deeply the relationship between the variables of a domain, favoring the increase of the comprehensibility of the discovered patterns, as well as the identification of the usefulness and relevance of these patterns.

In this paper, a new technique for modeling the graphical structure of a Bayesian network was presented, using multiple regressions as the method for analyzing the correlations among the attributes. The algorithm proposed implements a method for the structure learning which quantifies, based on mathematical models of regression, the level of the existing dependence from the variables. Initiating with a network without arcs and adding them in accordance with the identified correlations, as the search space of possible existing structures for the network is covered.

The tests carried to study the performance of the algorithm presented promising results in the observed aspects: regarding the quality of the generated network, when comparing with other learning algorithms, obtaining a representative structure of the Bayesian network, even when a reduced amount of data is available and; regarding its execution performance, achieving better execution times for bases with increasing volumes of data and, particularly, for its method of treating the attributes, discrete or continuous, not having its performance compromised as the number of possible states of the variables increases.

## References

1. Huang, H., Song, H., Tian, F., Lu, Y., e Wang, Q.: A comparatively research in incremental learning of Bayesian networks. *Intelligent Control and Automation. Fifth World Congress on 5*, 4260–4264 (2004)
2. Morales, M.M., Dominguez, R.G., Ramirez, N.C., Hernandez, A.G., e Andrade, J.L.J.: A method based on genetic algorithms and fuzzy logic to induce Bayesian networks, *Computer Science, 2004*. In: ENC 2004. Proceedings of the Fifth Mexican International Conference, pp. 176–180 (2004)
3. Larrañaga, P.: Structure Learning of Bayesian Networks by Genetic Algorithms: A Performance Analysis of Control Parameters. *IEEE Journal on Pattern Analysis and Machine Intelligence* 18(9), 912–926 (1996)
4. Rissanen, J.: Modeling by shortest data description. *Automatica* 14, 465–471 (1978)

5. Akaike, H.: A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19(6), 716–723 (1974)
6. Morales, M.M., Ramírez, N.C., Andrade, J.L.J., e Domínguez, R.G.: Bayes-N: an algorithm for learning Bayesian networks from data using local measures of information gain applied to classification problems. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) *MICAI 2004. LNCS (LNAI)*, vol. 2972, Springer, Heidelberg (2004)
7. Ramírez, N.C.: Building Bayesian networks from data: a constraint based approach, PhD Thesis, University of Sheffield (2001)
8. Spirtes, P.R., Shcheines, R., e Clark, G.: *TETRAD II: Tools for Discovery*. Lawrence Erlbaum Associates, Hillsdale, NJ., USA (1994)
9. Cooper, G., e Herskovitz, E.: A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning* 9, 309–347 (1992)
10. Shetty, S., Song, M.: Structure learning of Bayesian networks using a semantic genetic algorithm-based approach. In: *ITRE 2005. 3rd International Conference on Information Technology: Research and Education*, pp. 454–458 (2005)
11. Li, G., Tong, F., Dai, H.: Evolutionary structure learning algorithm for Bayesian network and Penalized Mutual Information metric, *Data Mining*. In: *ICDM 2001. Proceedings IEEE International Conference*, pp. 615–616 (2001)
12. Li, X.-L., Yuan, S.-M., He, X.-D.: Learning Bayesian networks structures based on extending evolutionary programming, In: *Machine Learning and Cybernetics. Proceedings of 2004 International Conference*, vol. 3, pp. 1594–1598 (2004)
13. Gamez, J.A., de Campos, L.M., Moral, S.: Partial abductive inference in Bayesian belief networks - an evolutionary computation approach by using problem-specific genetic operators. *Evolutionary Computation, IEEE Transactions* 6(2), 105–131 (2002)
14. Handa, H., Katai, O.: Estimation of Bayesian network algorithm with GA searching for better network structure. In: *Neural Networks and Signal Processing. Proceedings of the 2003 International Conference*, vol. 1, pp. 436–439 (2003)
15. Peng, H., Ding, C.: Structure search and stability enhancement of Bayesian networks, *Data Mining*. In: *ICDM 2003. Third IEEE International Conference*, pp. 621–624. IEEE Computer Society Press, Los Alamitos (2003)
16. Chickering, D.M., Heckerman, D., Meek, C.: Large-Sample Learning of Bayesian Networks is NP-Hard. *Journal of Machine Learning Research* 5, 1287–1330 (2004)
17. Cheng, J., Bell, D., e Liu, W.: Learning Bayesian Networks from Data: An Efficient Approach Based on Information Theory. *Artificial Intelligence*. 137(1-2), 43–90 (2002)
18. Robinson, R.W.: Counting unlabeled acyclic digraphs. In: *Proceedings of the Fifth Australian Conference on Combinatorial Mathematics*, pp. 28–43 (1976)
19. Herskovits, E.: *Computer-Based Probabilistic Networks Construction*, Ph.D. Thesis, Medical Information Sciences, University of Pittsburgh (1991)
20. Hair, J.F.J., Anderson, R.E., Tatham, R.L., e Black, W.C.: *Multivariate data analysis*. Prentice-Hall, Englewood Cliffs (1998)
21. Rice, J.A.: *Mathematical Statistics and Data Analysis*, 2nd edn. Duxbury Press, Boston, MA (1995)
22. Santana, A.C.: *Quantitative Methods in Econometry*, UFRA (2003)
23. Lauritzen, S.L., e Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. *Royal Statistics Society B50(2)*, 157–194 (1988)
24. Yang, S., e Chang, K.: Comparison of Score Metrics for Bayesian Network Learning. *IEEE Transactions on Systems, Man, and Cybernetics Part A32*, 419–428 (2002)

# Coordinating Returns Policies and Marketing Plans for Profit Optimization in E-Business Based on a Hybrid Data Mining Process

Chien-Chih Yu

Dept. Of Management Information Systems,  
National ChengChi University, Taipei, Taiwan  
ccyu@mis.nccu.edu.tw

**Abstract.** In electronic retailing market and/or supply chain, upstream sellers need to carefully plan and offer suitable returns policies to their downstream customers in order for ensuring product qualities, promoting product sales, maintaining customer satisfaction, lowering handling costs, and ultimately maximizing total profits. This paper aims at exploring the feasibility of using a hybrid data mining approach to support the coordination of returns policies and marketing plans for profit optimization in the e-business domain. A multi-dimensional data model and an integrated data mining process are provided to facilitate the three-staged clustering, classification, and association mining functions. Through this knowledge discovery process, customer and product classes identified in terms of the level of returns ratios are associated with the returns policies and marketing plans to generate rules for optimizing market profits. Also presented is a simulation example with embedded scenarios to test and validate the proposed data model and data mining process.

## 1 Introduction

A returns policy reflects the commitment of upstream sellers in the supply chain to accept excess or returned products from downstream buyers [14,18]. By carefully planning and offering suitable returns policies to their downstream retailers or customers, manufacturers or distributors may be able to ensure product qualities, promote product sales, maintain customer satisfaction, reduce handling costs, and ultimately maximize total profits. Many online as well as brick-and-mortar businesses including those in the computer, publication, food, and pharmaceutical industries have adopted and implemented various types of returns policies to attract customers and to build trust in buyer-seller relationships [3,4,6,11,15]. In the e-business domain characterized by more customized products, online retail stores, and direct sale channels, the returns policy has quickly emerged as an even more important strategic action for e-companies to sustain competitive advantages and profitability. Generally speaking, the most generous (loose) returns policy offers unconditional refund of wholesale/retail price for excess/returned products, on the contrary, the less generous (tighter) returns policy accepts no returns at all, offers no refund for or imposes some

types of restrictions on returned products [5,9,13,14]. It has been noted that a loose returns policy can entice customers' purchase interests and transactions to increase sales quantity, nevertheless, it can also increase the percentage of returns transactions and thus result in incurring more handling and logistic costs. In the research literature, issues related to customer-centric returns policy have been indicated as critical yet controversial for planning supply chain and marketing strategies [1,2,10,12,16,18]. However, most previous research works regarding returns policies still focus on upstream manufacturer-retailer interactions. These efforts mainly try to solve the policy planning problem as an optimization one by formulating a mathematical model in which sales profits is the objective function and the buyback price for returned products is the major decision variable. For examples, Padmanabhan and Png (1997) investigate the effect of a returns policy on pricing and stocking in the retailing business, and conclude that manufacturers should accept returns more loosely if production costs are sufficiently low and demand uncertainty is not too great [15]. Focusing on maximizing the supply chain joint profit, Lee (2001) analyzes the inter-organizational coordination effect in stocking, markdown sales, and returns policy for a distribution channel [10]. Choi, Li, and Yan (2004) address the issue of designing an optimal returns policy for a supply chain that is associated with a e-marketplace in which returned products can be sold with a higher price [2]. Hahn, Hwang, and Shinn (2004), instead, discuss the causal relationship between the supplier's discount price and the retailer's acceptance of no-returns policy when the coordination of perishable items in a distribution channel is the main concern [5]. Jointly considering the level of returns policy and the level of modularity in build-to-order product design, Mukhopadhyay and Setoputro (2005) present an approach for developing a profit maximization model for manufacturers [13]. Besides, Yao et al. (2005) estimate optimal order quantities for retailers and buyback prices for manufacturers in an environment having both retail and direct channels [18]. It can be seen that although these optimization approaches concerning returns policies of the manufacturer-retailer level do provide partial solutions for supply chain based strategic planning problems, many key factors related to the customer aspect are still missing in setting up a model for sufficiently representing returns policy decisions. To be more specific, neither customer and product characteristics nor purchase and returns transactions have been taken into account for designing and adopting customer-oriented returns policies. Within the customer relationship management (CRM) and supply chain management (SCM) context, demographic and transaction-based data of customers such as gender, age, education level, income level, frequency of purchase, average monetary of buying transactions, as well as buyers' returns patterns are all critical and useful for classifying customers, and for subsequently assigning returns policies to different customer classes aiming at reducing returns transactions and costs. However, in practice, the precious assets of customer and transaction based information have not been taken into account during the returns policy planning process. Furthermore, issues regarding returns-related influential factors as well as their joint effects, such as customer classes, product classes, returns policies, and marketing plans on returns patterns, are seldom addressed and desire for an in-depth exploration. Therefore, in order to make right decisions for adopting suitable returns policies, multiple factors from customers, products, transactions, and marketing dimensions must be jointly considered. As e-business sellers capture more knowledge about returns patterns of



customers and products, they can design and adopt better returns policies for specific customer classes associate with specific product categories to not only increase product sales but also decrease handling costs of returns transactions. It is expected that adopting suitable returns policies can eventually generate a win-win strategic outcome that benefits both buyers and sellers of the market and the supply chain.

The goal of this paper is to explore the potential of a hybrid data mining approach to support the planning and adoption of returns policies for e-business customers. At first, a multi-dimensional data framework is presented to illustrate key factors of returns transactions and policies. Subsequently, a hybrid data mining process is provided to direct a sequence of data mining activities including the clustering and segmentation of customers and products based on returns patterns, the classification of customer and product segments by returns ratios, as well as the association of customer and product classes with returns policies and pricing/promotion plans. In the following sections, section 2 and section 3 present respectively the integrated data framework for analyzing returns patterns and the hybrid data mining process for developing returns policies. The data mining and knowledge discovery process using an example with simulated data and embedded scenarios is demonstrated in Section 4 to validate the feasibility of the proposed framework and process. The final section contains a conclusion and future research directions.

## 2 The Integrated Data Framework

As aforementioned, for fully analyzing returns patterns, transaction data and associated data from the customer, product, returns policy, and marketing strategy (basically pricing and promotion) dimensions need to be taken into account. For the customer dimension, customers can be characterized by levels of the supply chain and demographic data. Levels of the supply chain including manufacturers, distributors, retailers, and individual buyers can be specified in terms of customer ID. Downstream buyers are customers of their upstream sellers. Elements of the customer-related demographic data include gender, age, education level, and income level, etc. Low, medium, and high can be used as the value range for age, education level, and income level. For the product dimension, major data elements include product name, type, price, size, ease of operation, and level of customization. Product types can be categorized as seasonal products, perishable items, digital and computer equipments, jewelry and highly expensive products, personalized and customized products, and many others. Similarly, low/medium/high and small/medium/large can be set as the value ranges for data elements: price, ease of operation, customization, and size levels. Build-to-order (customized) products, for instance, have the high level of customization. For the returns policy dimension, terms and restrictions are key attributes for characterizing the way to handle returns transactions. Types/levels of returns policies include loose, partial, and tight. The loose returns policy offers unconditional money back guarantee, while the tight returns policy accepts no refund whatsoever. In between, the partial returns policy provides product exchange, store credit for future purchase, or discounted refund price for returned products. So, values of the terms data element include no returns, repaired warrant, product exchange,

store credit, partial refund, and full money back guarantee. In addition, the restrictions element include unused product only, time limit for returning, return in original package, and other return instructions. The lower commitment in terms and higher complexity on restrictions imply tighter level of returns policies. For the marketing strategy dimension (mainly promotion plans), key attributes include start and end dates of the promotion period, and promotion types such as buy-one-get-one-free, double credits, and price discount, etc. As for the channel and time dimensions, types of selling channels include direct channel, e-marketplace, department store chain, specialty store chain, single outlet specialty stores, and auction site, while time related elements include year, month, date, and time of transactions (purchase or returns). A purchase transaction record contains information about monetary amount, returns policy, promotion plan, time, channel, and associated product line details (product, quantity, and subtotal). A returns transaction record, on the other hand, indicates returned product, quantity, money amount, time, and the original purchase transaction.

For detecting customers' returns patterns on purchased products, returns transactions are monitored and analyzed. Recency (R), frequency (F), and monetary (M) of customers' purchase and returns transactions (denoted as RFM-P and RFM-R respectively) are then derived for measurement and analysis. Within a specified time period for collecting and analyzing data, the recency of returns is the number of days from the latest returns transaction to the current day. The lower the result indicates that the customer has returns transaction more recently. The frequency of returns is the ratio of returns transactions and total purchase transactions within the span of the specified time horizon. The monetary of returns is the total refund/buyback prices of returned products for the given time period. Values of both the RFM-P and RFM-R related data elements are transformed into high (H), medium (M), and low (L) levels based on some pre-specified rules. In summary, using the purchase and returns transaction as the facts and the customers, products, returns policies, and marketing plans as the associated dimensions, an integrated multi-dimensional data framework for structurally representing factors of returns patterns, and associated relational tables in a simple form are presented in Figure 1.

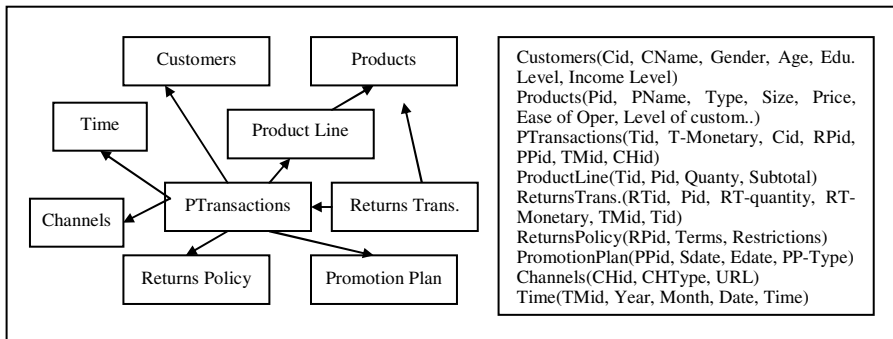


Fig. 1. The multi-dimensional data framework for analyzing returns patterns

### 3 The Integrated Data Mining Process

The integrated data mining process for supporting the analysis of returns patterns and the adoption of returns policies includes three stages that are clustering and segmentation, classification, and association rule generation respectively.

#### 3.1 Stage 1: The Clustering and Segmentation Process

In the clustering and segmentation stage, single dimensional clustering analyses are conducted regarding returns transactions with respect to the customer and the product dimensions. Algorithms such as the self-organizing maps (SOMs) and the k-means can be applied to determine the initial and final sets of clusters. For the customer dimension, the recency, frequency, and monetary of returns transactions (RFM-R) are derived as input data elements for clustering analysis. There will be totally less than 27 clusters to be generated within which the value set of the RFM-R for each cluster is a combination of high, medium, or low values such as (HHH), (MHL), or (LLH). For the product dimension, the type, price level, size level, and level of the ease of operation related to returned products are derived as the input data set.

Resulting clusters of both the customer and product dimensions are then segmented by means of critical data elements selected from the purchase and returns transactions as well as the customer and product dimensions. For instance, in the customer dimension, data of the derived returns ratios (RR), the derived RFM-P, and descriptive statistics related to customers' demographic attributes are used as the segmentation criteria. In this case, RR is defined as the ratio of the total returned items to the total purchased items of a customer within the specified time period. In the subsequent step, appropriate labels are given to the generated segments to reflect the returns patterns and potential profitability of these segmented customers. For an example, the clusters with H in RR, M to H in elements of the RFM-P, and H in income level are grouped into a segment that indicates a group of important customers with high returns to be watched. In other words, clusters with value sets including (H, HHH, H), (H, HMM, H), (H, MMH, H), and the like for (RR, RFM-P, Income L) are put into a segment labeled as high-class-high-returns. Similarly, segment with low-class-low-returns label can be identified. For the product dimension, the RR and price level elements can be chosen as the criteria for performing the segmentation process. In the product perspective, RR is the ratio of the total returned items to the total purchased items of a specific product within the specified time period. The value set (H, H) for (RR, Price L) indicates a high-price-high-returns segment that may incur high returns handling and restocking costs and thus needs to be further investigated.

At this stage, initial returns policies and marketing plans can be placed to the identified segments by applying a set of general rules, e.g. higher returns-tighter returns policy and higher profitability-looser returns policy, to increase sales and decrease returns handling costs. For instance, a tight returns policy is assigned to high-returns product and customer segments. But for the high-class-high-returns customer segment, a looser returns policy and a more favorable promotion plan are adopted.

### 3.2 Stage 2: The Classification Process

In the classification stage, the decision tree or other classification algorithms are used to derive classification rules that assess returns patterns of the product and customer classes. For the customer dimension, data elements including gender, age, education level, income level, and RR are used as the input data set for running the classification algorithm with RR as the labeling attribute. Using the generated classification rules, customers can be classified according to their demographic data to predict their returns patterns. More suitable returns policies and marketing plans for different customer classes can be applied to refine the initial ones for generating more sales while maintaining low potential returns ratios. New customers can be classified using these rules and connected to proper returns policies and promotion plans. For instance, if H in education level and M in age level imply high RR, then a tighter returns policy is adopted for this group of customers in the high-class-high-returns customer segment.

For the product dimension, data set containing type, price level, size level, ease of operation, and RR is used as input for classification. Similarly, a set of rules can be obtained for product classifications. Better returns policies as well as promotion plans can be chosen for specific product classes to improve the returns ratios.

### 3.3 Stage 3: The Association Mining Process

In stage 3, the association mining process is conducted. Based on purchase and returns transactions, customer class, product class, returns policy, marketing plan, and RR are chosen to form the input data set. The transaction-based RR is the ratio of the total returned items to the total purchased items of related purchase-returns transactions within the specified time period. All or pair-wised dimensional data elements can be used for running the association mining process to discover possible associations that may significantly influence the returns ratios. In other words, the impacts on returns across customer, product, returns policy, and marketing plan dimensions are examined in this stage. Through these cross-dimensional analyses, association rules with respect to customer and product classes, returns policy and marketing plan, as well as returns ratios can be generated. The classification scheme and the assignment of returns policies and promotion plans are then adjusted to get the final coordinated returns and marketing policies for specific customer and/or product classes. The major attempt of this approach is to derive optimal returns policies and marketing plans to increase the monetary of transactions as well as to decrease the returns ratios and costs. Some further findings can also be expected by carefully probing the association rules.

The entire integrated data mining process is shown in Figure 2.

## 4 An Example with Simulated Data and Embedded Scenarios

To validate the proposed integrated data mining approach, a set of simulated data with embedded scenarios [19], and a simplified returns policy-promotion plan coordination model are generated and tested. In the generated data set, there are totally 1000 purchase transactions with 100 returns transactions based on 100 customers and 50

products. Three major product types include jewelry and accessories (J), computers and electronics (C), and kitchen and housewares (K). Two test scenarios are set with respect to the customer and product dimensions. For the customer dimension, the scenario is that the younger female customers with high education level have high returns ratios. Using this simulated dataset with assumed returns patterns, the integrated data mining process for selecting coordinated returns policies and marketing plans is conducted in three stages. A simplified numerical example is also demonstrated to show the effect.

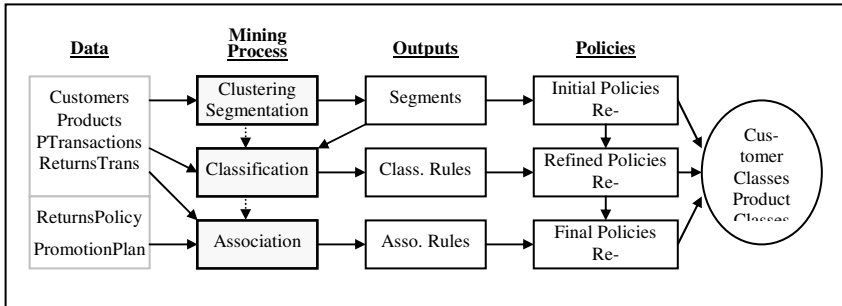


Fig. 2. The integrated data mining process

#### 4.1 The Data Mining Processes

In the clustering and segmentation stage, the derived RFM-R data set is used to perform the customer clustering analysis. Four customer clusters are obtained. These clusters are then examined by means of RR, RFM-P, and customers’ demographic data including gender, age, education level, and income level. As a result, three resulting segments are derived and labeled. Similarly, three product segments are obtained by analyzing the input data set that contains type, price level, size, ease of operation, and RR. The descriptive data as well as initial returns policies and promotion plans for these segmented classes are provided in Table 1. Returns policies are categorized into tight, partial, and loose, while promotion plans include buy-one-get-one-free (BIG1), double credit, and price discount. General rules for assigning initial returns policies include higher returns-tighter returns policy and higher profitability-looser returns policy as mentioned in the previous section. The notation H/M means that, in the segment, the percentage of H is greater than the percentage of M and the zero or small percentage of L is ignored.

In the subsequent classification mining stage, two decision trees and associated classification rules are generated for the customer and product dimensions. Totally, nine classification rules for the customer dimension and nine classification rules for the product dimension have been derived. The embedded scenario assuming that the high returns ratios for younger female customers with high education level has been successfully spotted in the segmented classes and associated classification rules. Another scenario for the product dimension .assuming that the high returns ratios for high-price and small-size products has also been tested. Two of these generated rules are “If the Gender is Female, and the Age level is Medium, and the Education Level

is High, then the Returns Ratio is High”, and “If the Price level is High, and the Size level is Low, then the Returns Ratio is High”. Figure 3 shows the result of a generated decision tree that classifies customers. Using the first customer-related classification rule, the tight returns policy can be assigned to the subclass of Customer-S1 with (F, M, H) value for (Gender, Age, Education) data. Similarly, the partial and loose returns policies can be adopted for Customer-S1, -S2 classes respectively. With this adjustment, more suitable returns policies for increasing sales and decreasing potential returns can be expected. Similar approaches for refining initial policies can be applied to the product classes.

**Table 1.** Customer and product segments with initial policies

Customer Segments	Attributes: (RFM-R, RFM-P, RR), and (Gender, Age, Education, Income)	Returns Policies	Promotion Plans
Customer-S1	(HHH, HMM, H), (F, M, H, H)	Tight, Partial	B1G1, Price discount
Customer-S2	(LMM, LMM, M), (F/M, M, M/H, M)	Partial	Price discount
Customer-S3	(LLL, LLM, L), (F/M, M/H, M/L, M/L)	Loose	Double credit
Product Segments	Attributes: (RR, Type, Price, Size, EaseofOperation)	Returns Policies	Promotion Plans
Product-S1	(H, J/C, H/M, L/M, H/M)	Tight	D. credit, P. discount
Product-S2	(M, C/K, M/H, H/M, M/H/L)	Partial	D. credit, P. discount
Product-S3	(L, K/C, M/H/L, H/M, L/H)	Loose	D. credit, P. discount

In the final association rule mining stage, more than twenty association rules are generated. Three significant rules for example include “If the Customer Class is Customer-S1 and the Product Class is Product-S1, then the Returns Ratio is High”, “If the Customer Class is Customer-S1 and the Product Class is Product-S2, then the Returns Ratio is Low”, and “If the Product Class is Product-S2 and the Promotion Plan is Double Credit, then the Returns Ratio is Low”. Table 2 lists these example association rules. These results can be used to finalize the returns policies and promotion plans for the customer and product classes. For instance, we can offer Customer-S1-(F, M, H) subclass the tight returns policy for Product-S1 and partial returns policy for Product-S2 products. For other customer classes buying products in Product-S2 class, a Double Credit promotion plan and a looser returns policy can be chosen as the final policies. One interesting finding is that the influence of promotion plans is not significant in previous stages, but the impact reveals when associated with specific product classes. To step forward, the final coordinated returns policy-marketing plan for Customer-S1and Product-S2 segment (denoted as C1-P2) can be set to (partial, double credit), and recorded as C1-P2(P,DC). In the same manner, we can also pick C1-P1(T,PD) that indicates assigning (tight, price discount) coordinated policy for the Customer-S1 and Product-S1 segment.

**4.2 A Simplified Model and Numerical Example**

For each product  $i$ , we assume that the decision variables of selling price and return price are denoted as  $p_i$  and  $r_i$ , and the parameters indicating unit cost and return handling cost are denoted as  $c_i$  and  $h_i$ . The selling quantity is denoted as  $Q_i$ , and may

be a function of  $p_i$  and  $r_i$ . Similarly, the return quantity is denoted as  $R_i$ , and may be a function of  $r_i$ . Consequently, the profit gained from selling product  $i$ , denoted as  $F_i$ , is a function of  $p_i$  and  $r_i$ , and can be computed by:  $F_i = (p_i - c_i)Q_i - (r_i + h_i)R_i$ . The objective is then to maximize the sum of  $F_i$  by choosing  $p_i$  and  $r_i$  for all  $i$ .

```

=== Run information ===
Instances: 100
Attributes: 5
      Gender, Edu, Age, Income, Returns Ratio
Test mode: evaluate on training data
=== Classifier model (full training set) ===
J48 pruned tree
-----
Edu = H
| Gender = F
| | Age = M: RR=H (13.0)
| | Age = H: RR=M (2.0)
| | Age = L: RR=H (0.0)
| Gender = M: RR=L (10.0)
Edu = M/H
| Age = M: RR=M (21.0/4.0)
| Age = H: RR=L (7.0)
| Age = L: RR=M (0.0)
Edu = M: RR=L (27.0)
Edu = L: RR=L (20.0)

Number of Leaves : 9
Size of the tree : 13
=== Summary ===
Correctly Classified Instances      96      96  %
Incorrectly Classified Instances    4        4  %
Kappa statistic                    0.9207
Mean absolute error                 0.0432
Root mean squared error             0.1469
Relative absolute error             13.2193 %
Root relative squared error         36.551 %
    
```

**Fig. 3.** A decision tree for customer classification

**Table 2.** List of three example association rules

IF	Then
Customer Class S1 and Product Class S1	Returns Ratio=High, Support: 0.35 Conf:0.75
Customer Class S1 and Product Class S2	Returns Ratio=Low, Support: 0.37 Conf: 0.7
Product Class S2 and Promotion plan D. credit	Returns Ratio=Low, Support: 0.32 Conf: 0.75

In addition, for obtaining a theoretical solution,  $Q_i$  and  $R_i$  need to be fitted as functions of  $(p_i, r_i)$  and  $r_i$  respectively in a way that  $F_i$  is concave. However, this could be difficult in real case since customers' buying behavior and return patterns with respect to different products or product classes may vary in a great deal.

For showing the effect on profit by coordinating returns policies and price promotion plans, we simplify the problem by setting  $i=1$ ,  $c=6$ ,  $h=2$ , and the initial  $p=10$ , and  $r=10$ . As the example given in the previous subsection 4.1,  $Q=1000$ ,  $R=100$ , and the current profit  $F$  is 2800. By using 16%, 12%, and 8% as the high, medium, and low return rates respectively, the (no. purchase trans, no. returns trans) data for three derived customer classes Customer-S1, Customer-S2, and Customer-S3, each has 13, 23, and 64 customers, are then set to be (250,40), (250,30), and (500,30) respectively. To increase the profit with  $p=10$  being fixed, we can change the original

full buyback price returns policy, i.e.  $r=10$ , to a combined 0, 6, 10 of return prices for high, medium, and low returns-rate customer classes C1, C2, and C3 respectively, and gain a better total profit  $F=3320$ . We write the coordinated Customer(selling, returns) pricing policies, as [C1(10,0),C2(10,6),C3(10,10),  $F=3320$ ]. Other possible sets are as [C1(10,0),C2(9,6),C3(10,10),  $F=3070$ ] or [C1(9,0),C2(10,6),C3(10,10),  $F=3070$ ], etc. Table 3 shows the numerical result of the first set coordinated policies. When both customer and product classes are involved, the coordinated policy looks like [Ci-Pj( $p_{ij}$ ,  $r_{ij}$ ),  $F$ ], where ( $p_{ij}$ ,  $r_{ij}$ ) indicating (selling price, returns price) for the Ci-Pj segment.

**Table 3.** Result of a coordinated selling/returns pricing policies

C-class	No. P-trans.	No. R-trans.	S-price	R-price	Ci Profits	Total profit F
C1	250	40	10	0	920	
C2	250	30	10	6	760	
C3	500	30	10	10	1640	3320

## 5 Conclusion

Returns policy has been noted as an important competitive weapon in the e-market to generate more profits. Popular web-based e-shop and e-auction sites such as Amazon, Yahoo, and e-Bay all provide on-line access to their returns policies. However, how to adopt proper returns policies for customer and product classes has still been considered as a crucial yet complex issue for e-business to sustain high profitability. In this paper, we present a multi-dimensional data framework and an integrated data mining process to deal with the adoption and coordination of returns policies and marketing plans in a systematic way. The three-stage data mining process includes the clustering and segmentation stage, the classification mining stage, and the association rule generation stage. Through these stages, customers and products are clustered and segmented, classification rules are generated for the segmented and labeled classes, and association rules are derived across multiple dimensions including customer, product, returns policy, and promotion plan. By using the segmented classes, classification rules, and association rules, coordinated returns policies and promotion plans can be initially assigned, subsequently refined, and finally decided and adopted for labeled customer and product classes to leverage profit gains. A simplified example is tested to show the effect of the proposed integrated data mining approach.

Since customer and transaction based e-market information has been frequently ignored during the returns policy planning and control process, no real data covering all required and related data dimensions are available for conducting real-case experiments. It is expected that this research will attract and guide e-market participants to realize more about the returns patterns, and to start collecting data based upon the proposed framework and process for planning better returns policies. Future research works will include extending the proposed data framework and data mining process to the entire e-supply chain for optimizing the global chain profit, and conducting real world experiments to validate the effectiveness of this approach.



## References

1. Arya, A., Mittendorf, B.: Using Return Policies to Elicit Retailer Information. *Rand Journal of Economics* 35(3), 617–630 (2004)
2. Choi, T.M., Li, D., Yan, H.: Optimal Returns Policy for Supply Chain with e-Marketplace. *International Journal of Production Economics* 88(2), 205–227 (2004)
3. Davis, S., Hagerty, M., Gerstner, E.: Return Policies and the Optimal Level of Hassle. *Journal of Economics and Business* 50(5), 445–460 (1998)
4. Eduardo, R.O., Andres, R.P.: The Regional Return of Public Investment Policies in Mexico. *World Development* 32(9), 1545–1562 (2004)
5. Hahn, K.H., Hwang, H., Shinn, S.W.: A Returns Policy for Distribution Channel Coordination of Perishable Items. *European Journal of Operational Research* 152(3), 770–780 (2004)
6. Hoffman, W., Keedy, J., Roberts, K.: The Unexpected Return of B2B. *The McKinsey Quarterly* (3), 97–105 (2002)
7. Hsieh, N.C.: Hybrid Mining Approach in the Design of Credit Scoring Models. *Expert Systems with Applications* 28(4), 655–665 (2005)
8. Kuo, R.J., Ho, L.M., Hu, C.M.: Integration of Self-Organizing Feature Map and K-means Algorithm for Marketing Segmentation. *Computer and Operations Research* 29(11), 1475–1493 (2002)
9. Lau, A.H.L., Lau, H.S., Willett, D.K.: Demand Uncertainty and Returns Policies for a Seasonal Product: An Alternative Model. *International Journal of Production Economics* 66(1), 1–12 (2000)
10. Lee, C.H.: Coordinated Stocking, Clearance Sales, and Return Policies for a Supply Chain. *European Journal of Operational Research* 131(3), 491–513 (2001)
11. Longo, T.: At Stores, Many Unhappy Returns. *Kiplinger's Personal Finance Magazine* 49(6), 103 (1995)
12. Mantrala, M.K., Raman, K.: Demand Uncertainty and Supplier's Returns Policies for a Multi-Store Style-Good Retailer. *European Journal of Operational Research* 115(2), 270–284 (1999)
13. Mukhopadhyay, S.K., Setoputro, R.: Optimal Return Policy and Modular Design for Build-To-Order Products. *Journal of Operations Management* 23(5), 496–506 (2005)
14. Padmanabhan, V., Png, I.P.L.: Returns Policies: Make Money by Making Good. *Sloan Management Review* 37(1), 65–72 (1995)
15. Padmanabhan, V., Png, I.P.L.: Manufacturer's Returns Policies and Retail Competition. *Marketing Science* 16(1), 81–94 (1997)
16. Tsay, A.A.: Managing Retail Channel Overstock: Markdown Money and Return Policies. *Journal of Retailing* 77(4), 457–492 (2001)
17. Webster, S., Weng, Z.K.: A Risk-free Perishable Item Returns Policy. *Manufacturing & Service Operations Management* 2(1), 100–106 (2000)
18. Yao, D.Q., Yue, X., Wang, X., Liu, J.J.: The Impact of Information Sharing on a Returns Policy with the Addition of a Direct Channel. *International Journal of Production Economics* 97(2), 196–209 (2005)
19. Yu, C.C., Wang, C.S.: A Hybrid Mining Approach for Optimizing Returns Policies in eRetailing. In: *Proceedings of the 5th International Conference on Computational Intelligence in Economics and Finance*, pp. 38–41 (2006)

# An EM Algorithm to Learn Sequences in the Wavelet Domain

Diego H. Milone and Leandro E. Di Persia

Signals and Computational Intelligence Laboratory - CONICET  
Department of Informatics, Faculty of Engineering and Water Sciences  
National University of Litoral, Ciudad Universitaria, Santa Fe, Argentina  
{dmilone,ldipersia}@fich.unl.edu.ar  
<http://fich.unl.edu.ar/sinc>

**Abstract.** The wavelet transform has been used for feature extraction in many applications of pattern recognition. However, in general the learning algorithms are not designed taking into account the properties of the features obtained with discrete wavelet transform. In this work we propose a Markovian model to classify sequences of frames in the wavelet domain. The architecture is a composite of an external hidden Markov model in which the observation probabilities are provided by a set of hidden Markov trees. Training algorithms are developed for the composite model using the expectation-maximization framework. We also evaluate a novel delay-invariant representation to improve wavelet feature extraction for classification tasks. The proposed methods can be easily extended to model sequences of images. Here we present phoneme recognition experiments with TIMIT speech corpus. The robustness of the proposed architecture and learning method was tested by reducing the amount of training data to a few patterns. Recognition rates were better than those of hidden Markov models with observation densities based in Gaussian mixtures.

**Keywords:** EM algorithm, Hidden Markov Models, Hidden Markov Trees, Speech Recognition, Wavelets.

## 1 Introduction

Hidden Markov trees (HMT) have been recently introduced [1] to model the statistical dependencies at different scales and the non-Gaussian distributions of the wavelet coefficients [2]. In the last years, the HMT model was improved in several ways, for example, using more states within each HMT node and developing more efficient algorithms for initialization and training [3,4].

Discrete and continuous hidden Markov models (HMM) have been used in applications of machine learning and pattern recognition, such as computer vision, bioinformatics, speech recognition, medical diagnosis and many others [5,6,7,8]. A well-known model for continuous observation densities is the Gaussian mixture model (GMM) [9]. The HMM-GMM architecture is a widely used model, for example, in speech recognition [10]. Nevertheless, more accurate models have

been proposed for the observation densities [11]. In both, discrete and continuous models, the most important advantage of the HMM lies in that they can deal with sequences of variable length. However, if the whole sequence is analyzed by the discrete wavelet transform (DWT), like in the case of HMT, a representation whose structure is dependent on the sequence length is obtained. Therefore, the learning architecture should be trained and used only for this sequence length. On the other hand, in HMM modeling, stationarity is generally assumed withing each observation in the sequence. This stationarity assumption can be removed when observed features are extracted by the DWT, but a suitable statistical model for learning this features in the wavelet domain would be needed.

Fine et al. [12] proposed a recursive hierarchical generalization of discrete HMM. They apply the model to learn the multiresolution structure of natural English text and cursive handwriting. Some years later, Murphy and Paskin [13] derived a simpler inference algorithm by formulating the hierarchical HMM as a special kind of dynamic Bayesian network. A wide review about multiresolution Markov models was provided in [14], with special emphasis on applications to signal and image processing. Dasgupta et al. [15] proposed a dual-Markov architecture, trained by means of an iterative process where the most probable sequence of states is identified, and then each internal model is adapted with the selected observations. A similar approach applied to image segmentation is proposed in [16]. However, in these cases the model consists of two separated and independent entities, that are forced to work in a coupled way. By the contrary, Bengio et al. derived a training algorithm for the full model [17], composed of an external HMM in which for each state an internal HMM provides the observation probability distribution [18]. In the following, we derive an EM algorithm for a composite HMM-HMT architecture that observes sequences of DWTs in  $\mathbb{R}^N$ . This algorithm can be easy generalized to sequences in  $\mathbb{R}^{N \times N}$  with 2-D HMTs like the used in [19] or [20].

In the next section we introduce the notation for HMM and HMT. Using this notation we review the training algorithms by defining the joint likelihood and then deriving the reestimation formulas. In Section 3, the experimental results for speech recognition using real data are presented and discussed. Two different alternatives for the feature extraction with DWT are tested. The experiments are carried out by training the proposed model and the HMM-GMM with a variable amount of data to test their robustness with a few training patterns. In the last section we present the main conclusions and some ideas for future works.

## 2 The HMM-HMT Model

The architecture proposed in this work is a composition of two Markov models: the long term dependencies are modeled with an external HMM and each pattern in the local context is modeled with an HMT.

## 2.1 Basic Definitions

To model a sequence  $\mathbf{W} = \mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^T$ , with  $\mathbf{w}^t \in \mathbb{R}^N$ , a continuous HMM is defined with the structure  $\vartheta = \langle \mathcal{Q}, \mathbf{A}, \boldsymbol{\pi}, \mathcal{B} \rangle$ , where:

- i)  $\mathcal{Q} = \{Q \in [1 \dots N_Q]\}$  is the set of states.
- ii)  $\mathbf{A} = [a_{ij} = \Pr(Q^t = j | Q^{t-1} = i)]$ ,  $\forall i, j \in \mathcal{Q}$ , is the matrix of transition probabilities, where  $Q^t \in \mathcal{Q}$  is the model state at time  $t \in [1 \dots T]$ ,  $a_{ij} \geq 0 \forall i, j$  and  $\sum_j a_{ij} \doteq 1 \forall i$ .
- iii)  $\boldsymbol{\pi} = [\pi_j = \Pr(Q^1 = j)]$  is the initial state probability vector. In the case of left-to-right HMM this vector is  $\boldsymbol{\pi} = \boldsymbol{\delta}_1$ .
- iv)  $\mathcal{B} = \{b_k(\mathbf{w}^t) = \Pr(\mathbf{W}^t = \mathbf{w}^t | Q^t = k)\}$ ,  $\forall k \in \mathcal{Q}$ , is the set of observation (or emission) probability distributions.

Let be  $\mathbf{w} = [w_1, w_2, \dots, w_N]$  resulting of a DWT analysis with  $J$  scales and without including  $w_0$ , the approximation coefficient at the coarsest scale (that is,  $N = 2^J - 1$ ). The HMT can be defined with the structure  $\theta = \langle \mathcal{U}, \mathcal{R}, \boldsymbol{\pi}, \boldsymbol{\epsilon}, \mathcal{F} \rangle$ , where:

- i)  $\mathcal{U} = \{u \in [1 \dots N]\}$  is the set of nodes in the tree.
- ii)  $\mathcal{R} = \{R \in [1 \dots NM]\}$  is the set of states in all the nodes of the tree, denoting with  $\mathcal{R}_u = \{R_u \in [1 \dots M]\}$  the set of states in the node  $u$ .
- iii)  $\boldsymbol{\epsilon} = [\epsilon_{u,mn} = \Pr(R_u = m | R_{\rho(u)} = n)]$ ,  $\forall m \in \mathcal{R}_u, \forall n \in \mathcal{R}_{\rho(u)}$ , is the array whose elements hold the conditional probability of node  $u$  being in state  $m$  given that the state in its parent node  $\rho(u)$  is  $n$ , where  $\sum_m \epsilon_{u,mn} \doteq 1$ .
- iv)  $\boldsymbol{\pi} = [\pi_p = \Pr(R_1 = p)]$ ,  $\forall p \in \mathcal{R}_1$  are the probabilities for the root node being on state  $p$ .
- v)  $\mathcal{F} = \{f_{u,m}(w_u) = \Pr(W_u = w_u | R_u = m)\}$  are the observation probability distributions. This is,  $f_{u,m}(w_u)$  is the probability of observing the wavelet coefficient  $w_u$  with the state  $m$  (in the node  $u$ ).

In the following, we will simplify the notation for random variables. For example, we write  $\Pr(w_u | r_u)$  instead of  $\Pr(W_u = w_u | R_u = r_u)$ .

## 2.2 Joint Likelihood

Let be  $\Theta$  an HMM like the one defined above but using a set of HMTs to model the observation densities within each HMM state:

$$b_{q^t}(\mathbf{w}^t) = \sum_{\forall \mathbf{r}} \prod_{\forall u} \epsilon_{u,r_u r_{\rho(u)}}^{q^t} f_{u,r_u}^{q^t}(w_u^t), \quad (1)$$

with  $\mathbf{r} = [r_1, r_2, \dots, r_N]$  a combination of hidden states in the HMT nodes. To extend the notation in the composite model, we have added a superscript in the HMT variables to make reference to the state in the external HMM. For example,  $\epsilon_{u,mn}^k$  will be the conditional probability that, in the state  $k$  of the external HMM, the node  $u$  is in state  $m$  given that the state of its parent node  $\rho(u)$  is  $n$ .

Thus, the complete joint likelihood for the HMM-HMT can be obtained as

$$\begin{aligned} \mathcal{L}_\Theta(\mathbf{W}) &= \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \prod_t a_{q^{t-1}q^t} \prod_{\forall u} \epsilon_{u,r_u^t,r_{\rho(u)}^t}^{q^t} f_{u,r_u^t}^{q^t}(w_u^t) \\ &\triangleq \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_\Theta(\mathbf{W}, \mathbf{q}, \mathbf{R}), \end{aligned} \tag{2}$$

where we simplify  $a_{01} = \pi_1 = 1$ ,  $\forall \mathbf{q}$  is over all possible state sequences  $\mathbf{q} = q^1, q^2, \dots, q^T$  and  $\forall \mathbf{R}$  are all the possible sequences of all the possible combinations of hidden states  $\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^T$  in the nodes of each tree.

### 2.3 Training Formulas

In this section we will obtain the maximum likelihood estimation of the model parameters. For the optimization, the auxiliary function can be defined as

$$\mathcal{D}(\Theta, \bar{\Theta}) \triangleq \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_\Theta(\mathbf{W}, \mathbf{q}, \mathbf{R}) \log(\mathcal{L}_{\bar{\Theta}}(\mathbf{W}, \mathbf{q}, \mathbf{R})) \tag{3}$$

and using (2)

$$\begin{aligned} \mathcal{D}(\Theta, \bar{\Theta}) &= \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_\Theta(\mathbf{W}, \mathbf{q}, \mathbf{R}) \cdot \left\{ \sum_t \log(a_{q^{t-1}q^t}) + \right. \\ &\quad \left. + \sum_t \sum_{\forall u} \left[ \log\left(\epsilon_{u,r_u^t,r_{\rho(u)}^t}^{q^t}\right) + \log\left(f_{u,r_u^t}^{q^t}(w_u^t)\right) \right] \right\}. \end{aligned} \tag{4}$$

For the estimation of the transition probabilities in the HMM,  $a_{ij}$ , no changes from the standard formulas will be needed. However, on each internal HMT we hope that the estimation of the model parameters will be affected by the probability of being in the HMM state  $k$  at time  $t$ .

Let be  $q^t = k$ ,  $r_u^t = m$  and  $r_{\rho(u)}^t = n$ . To obtain the learning rule for  $\epsilon_{u,mn}^k$  the restriction  $\sum_m \epsilon_{u,mn}^k \triangleq 1$  should be satisfied. If we use

$$\hat{\mathcal{D}}(\Theta, \bar{\Theta}) \triangleq \mathcal{D}(\Theta, \bar{\Theta}) + \sum_n \lambda_n \left( \sum_m \epsilon_{u,mn}^k - 1 \right), \tag{5}$$

the learning rule results

$$\epsilon_{u,mn}^k = \frac{\sum_t \gamma^t(k) \xi_u^{tk}(m, n)}{\sum_t \gamma^t(k) \gamma_{\rho(u)}^{tk}(n)}, \tag{6}$$

where  $\gamma^t(k)$  is computed as usual for HMM and  $\gamma_{\rho(u)}^{tk}(n)$  and  $\xi_u^{tk}(m, n)$  can be estimated with the upward-downward algorithm [4].

For the observation distributions we use  $f_{u,r_u}^{q^t}(w_u^t) = \mathcal{N}(w_u^t, \mu_{u,r_u}^{q^t}, \sigma_{u,r_u}^{q^t})$ . From (4) we have

$$\begin{aligned} \mathcal{D}(\Theta, \bar{\Theta}) &= \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_{\Theta}(\mathbf{W}, \mathbf{q}, \mathbf{R}) \cdot \left[ \sum_t \log(a_{q^{t-1}q^t}) + \sum_t \sum_{\forall u} \log \left( \epsilon_{u,r_u}^{q^t} \right) + \right. \\ &\quad \left. + \sum_t \sum_{\forall u} \left( -\frac{\log(2\pi)}{2} - \log \left( \sigma_{u,r_u}^{q^t} \right) - \frac{\left( w_u^t - \mu_{u,r_u}^{q^t} \right)^2}{2 \left( \sigma_{u,r_u}^{q^t} \right)^2} \right) \right]. \end{aligned} \tag{7}$$

Thus, the training formulas result:

$$\mu_{u,m}^k = \frac{\sum_t \gamma^t(k) \gamma_u^{tk}(m) w_u^t}{\sum_t \gamma^t(k) \gamma_u^{tk}(m)} \quad \text{and} \quad (\sigma_{u,m}^k)^2 = \frac{\sum_t \gamma^t(k) \gamma_u^{tk}(m) (w_u^t - \mu_{u,m}^k)^2}{\sum_t \gamma^t(k) \gamma_u^{tk}(m)}.$$

### 2.4 Multiple Observation Sequences

In practical situations we have a training set with a large number of observed data  $\mathcal{W} = \{\mathbf{W}^1, \mathbf{W}^2, \dots, \mathbf{W}^P\}$ , where each observation consists of a sequence of evidences  $\mathbf{W}^p = \mathbf{w}^{p,1}, \mathbf{w}^{p,2}, \dots, \mathbf{w}^{p,T_p}$ , with  $\mathbf{w}^{p,t} \in \mathbb{R}^N$ . In this case we define the auxiliary function

$$\mathcal{D}(\Theta, \bar{\Theta}) \triangleq \sum_{p=1}^P \frac{1}{\Pr(\mathbf{W}^p|\theta)} \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_{\Theta}(\mathbf{W}^p, \mathbf{q}, \mathbf{R}) \log(\mathcal{L}_{\bar{\Theta}}(\mathbf{W}^p, \mathbf{q}, \mathbf{R})) \tag{8}$$

and replacing with (2)

$$\begin{aligned} \mathcal{D}(\Theta, \bar{\Theta}) &= \sum_{p=1}^P \frac{1}{\Pr(\mathbf{W}^p|\theta)} \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_{\Theta}(\mathbf{W}^p, \mathbf{q}, \mathbf{R}) \cdot \left\{ \sum_{t=1}^{T_p} \log(a_{q^{t-1}q^t}) + \right. \\ &\quad \left. + \sum_{t=1}^{T_p} \sum_{\forall u} \left[ \log \left( \epsilon_{u,r_u}^{q^t} \right) + \log \left( f_{u,r_u}^{q^t}(w_u^{p,t}) \right) \right] \right\}. \end{aligned} \tag{9}$$

The derivation of the training formulas is similar to the ones presented for a single sequence. We simply summarize here the main results:

$$a_{ij} = \frac{\sum_{p=1}^P \sum_{t=1}^{T_p} \xi^{p,t}(i, j)}{\sum_{p=1}^P \sum_{t=1}^{T_p} \gamma^{p,t}(i)}, \quad (\sigma_{u,m}^k)^2 = \frac{\sum_{p=1}^P \sum_{t=1}^{T_p} \gamma^{p,t}(k) \gamma_u^{p,tk}(m) (w_u^{p,t} - \mu_{u,m}^k)^2}{\sum_{p=1}^P \sum_{t=1}^{T_p} \gamma^{p,t}(k) \gamma_u^{p,tk}(m)},$$

$$\epsilon_{u,mn}^k = \frac{\sum_{p=1}^P \sum_{t=1}^{T_p} \gamma^{p,t}(k) \xi_u^{p,tk}(m, n)}{\sum_{p=1}^P \sum_{t=1}^{T_p} \gamma^{p,t}(k) \gamma_{\rho(u)}^{p,tk}(n)}, \mu_{u,m}^k = \frac{\sum_{p=1}^P \sum_{t=1}^{T_p} \gamma^{p,t}(k) \gamma_u^{p,tk}(m) w_u^{p,t}}{\sum_{p=1}^P \sum_{t=1}^{T_p} \gamma^{p,t}(k) \gamma_u^{p,tk}(m)}$$

### 3 Experimental Results and Discussion

In this section we test the proposed model in the context of automatic speech recognition with the TIMIT corpus [21]. In the following the data set for the experiments is briefly described. Subsection 3.2 details the first experiments with the standard DWT and comparing the performance of the proposed model and the others which are often used for this classification task. Next, we present and discuss the results obtained by introducing an improvement in the DWT feature extraction for speech data. In the Subsection 3.4 we present a set of tests aimed to evaluate the robustness of the models to the reduction of the amount of training data.

#### 3.1 Data Sets and Implementation Details

TIMIT is a well known corpus that has been used extensively for research in automatic speech recognition. From this corpus five phonemes that are difficult to classify were selected. The voiced stops /b/ and /d/ have a very similar articulation (bilabial/alveolar) and different phonetic variants according to the context (allophones). Vowels /eh/ and /ih/ were selected because their formants are very close. Thus, these phonemes are very confusable. To complete the selected phonemes, the affricate phoneme /jh/ was added as representative of the voiceless group [22]. Table 1 shows the number of train and test samples for each phoneme in all the dialectical regions of the TIMIT corpus.

**Table 1.** Selected phonemes from TIMIT speech corpus

Phonemes	/b/	/d/	/eh/	/ih/	/jh/
Train patterns	2181	3548	3853	5051	1209
Test patterns	886	1245	1440	1709	372

Regarding practical issues, the training formulas were implemented in logarithmic scale to make a more efficient computation of products and to avoid underflow errors in the probability accumulators [4]. In addition, underflow errors are reduced because in the HMM-HMT architecture each DWT is in a lower dimension than the dimension resulting from an unique HMT for the whole sequence (like in [1] and [4]). All learning algorithms and transforms used in the experiments were implemented in C++ from scratch.

### 3.2 Standard DWT Features

Frame by frame, each local feature is extracted using a Hamming window of width  $N_w$ , shifted in steps of  $N_s$  samples [10]. The first window begins  $N_o$  samples out (with zero padding) to avoid the information loss at the beginning of the sequence. The same procedure is used to avoid information loss at the end of the sequence. Then a DWT is applied to each windowed frame. The DWT was implemented by the fast pyramidal algorithm [23], using periodic convolutions and the Daubechies-8 wavelet [24]. Preliminary tests were carried out with other wavelets of the Daubechies and Splines families but not important differences in results were found.

In this first study, different recognition architectures are compared, but setting them to have the total number of trainable parameters in the same order of magnitude. A separate model is trained for each phoneme and the recognition is made by the conventional maximum-likelihood classifier. Table 2 shows the recognition rates (RR) for: GMM with 4 Gaussians in the mixture (2052 trainable parameters), HMT with 2 states per node and one Gaussian per state (2304 trainable parameters), HMM-GMM with 3 states and 4 Gaussians in each mixture (6165 trainable parameters), HMM-HMT with 3 HMM states, 2 states per HMT node and one Gaussian per node state (6921 trainable parameters) [1].

For HMM-GMM and HMM-HMT the external HMM have connections  $i \rightarrow i$ ,  $i \rightarrow (i + 1)$  and  $i \rightarrow (i + 2)$ . The last link allows to model the shortest sequences, with less frames than states in the model. In both, GMM and HMM-GMM, the Gaussians in the mixture are modeled with diagonal covariance matrices.

The maximum number of reestimations used for all experiments was 10, but also, as finalization criteria, the training process was stopped if the average (log) probability of the model given the training sequences was improved less than 1%. In most of the cases, the training converges after 4 to 6 iterations, but HMM-GMM models experienced several convergence problems with the DWT data. When a convergence problem was observed, the model corresponding to the last estimation with an improvement in the average probability of the model given the sequences was used for testing.

Results in Table 2 were obtained with the dialectical region 1 of the TIMIT corpus (1145 phonemes for train and 338 for test). Recognition rates (RR) for HMT are higher than those achieved by GMM, mainly because HMT provides a better model for the structure in the wavelet coefficients. However, the capability of the HMM-GMM to model the dynamics of the sequences of frames allows to improve RR over HMT. But moreover, the combination of the two advantages in the HMM-HMT surpass all previous results.

The computational cost may be one of the major handicaps of the proposed approach, mainly because of the double Baum-Welch process required in the training. To provide an idea of the computational cost, results reported in Table 2, with  $N_w = 256$  and  $N_s = 128$ , demand 30.20 s of training for the HMM-GMM whereas the same training set demands 240.89 s in the HMM-HMT [2].

<sup>1</sup> All these counts are for  $N_w = 256$ .

<sup>2</sup> Using a Intel Core 2 Duo E6600 processor.



**Table 2.** Recognition rates (RR%) for TIMIT phonemes (dialectal region 1) using models with a similar number of trainable parameters

Learning Architecture	Frame size $N_w$		Average RR%
	128	256	
GMM	28.99	29.88	29.44
HMT	31.36	36.39	33.88
HMM-GMM	35.21	37.87	36.54
HMM-HMT	47.34	39.64	43.49

### 3.3 Delay-Invariant DWT Features

The next experiments were aimed to the comparison of the two main models related with this work, that is, HMM using observation probabilities provided by GMMs or HMTs. In this context, the best relative scenario for HMM-GMM is using  $N_w = 256$  and  $N_s = 128$  (see Table 2).

Furthermore, in this section we will study a problem that arises in the feature extraction with the DWT, applied in the context of a frame by frame analysis. When a quasi-periodic waveform is analyzed by DWT, it can be seen that the major peak is replicated within each scale. In a frame by frame analysis, the positions of these peaks are related to the difference between the starting time of the frame and the location of the maximum. Thus, this is an artifact not related with the identity of the phoneme. Undoubtedly, these artifacts make training data too confusable for any recognition architecture without translation-invariance. A wavelet representation for quasi-periodic signals was proposed in [25]. Other authors integrate all the coefficients within each scale, using only the subband energy information of the DWT. Then, they apply a principal component analysis [26] or the cosine transform [27] to decorrelate the frame. However, a simpler idea can be applied to avoid this information loss: the spectrum module by scale (SMS) can be used rather than the wavelet coefficients themselves. This solution preserves the DWT information within scales without a major complexity in the implementation and remove the artifact generated by the quasi-periodic behavior of the waveform. In the following experiments we will refer to this method for feature extraction as SMS-DWT.

In Table 3 we present a fine tuning for the HMM-GMM model, with DWT and SMS-DWT feature extraction. Note that comparable architectures for HMM-HMT are HMM-GMM with between 2 to 8 Gaussians in the mixtures, because they have a similar number of trainable parameters. In spite of all the tested HMM-GMM alternatives, the HMM-HMT is still providing the best recognition rates. The improvement obtained by the SMS method is very important for both models. The proposed model is still providing the best results, mainly because the structural relations between the wavelet scales (modeled by the HMTs) are preserved after the SMS post-processing.

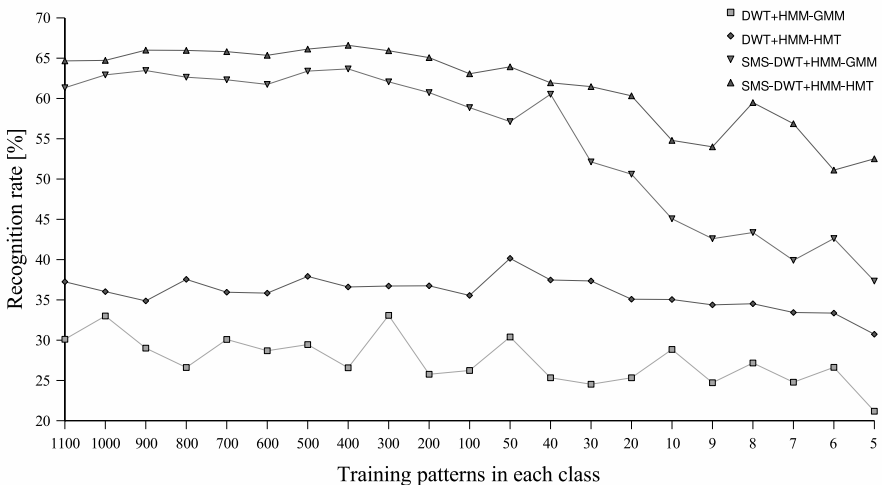
**Table 3.** Recognition results for TIMIT phonemes applying the DWT directly to each frame and with the SMS post-processing. All dialectical region of TIMIT corpus were used in these experiments. Note that HMM-HMT Gaussians are in  $\mathbb{R}^1$  whereas HMM-GMM Gaussians are in  $\mathbb{R}^{256}$ .

Learning Architecture	Gaussians per GMM	Total of parameters	Recognition rates [%]	
			DWT	SMS-DWT
HMM-GMM	2	3087	29.60	63.07
	4	6165	28.33	63.40
	8	12321	33.73	62.20
	16	24633	28.93	62.00
	32	49257	27.00	63.00
	64	98505	33.47	59.13
HMM-HMT	512	6921	37.93	66.27

### 3.4 Robustness to the Amount of Training Data

Fig. 1 shows the results when the training data is reduced to a few patterns. Each recognition rate on this figure is the average of 10 independent trials, with training patterns selected at random. From the testing set of TIMIT corpus, 300 patterns for each class were also selected at random for each trial.

This figure shows that, when there is a sufficient amount of training data, recognition rates grow up to the reported in Table 3. However, when the amount of training data is reduced, HMM-GMM performance is significantly more affected than the performance of HMM-HMT. In the case of SMS-DWT, the RR of HMM-GMM fall from 61.33% to 37.34% (about the same RR that the



**Fig. 1.** Performance of proposed architecture for classification using different amounts of training patterns

average for HMM-HMT with the standard DWT features). In contrast, RR for the HMM-HMT only falls from 64.67% to 52.51%.

On the other hand, it can be observed that with standard DWT features the performance of the HMM-GMM falls more slowly, from 30.11% to 21.18%. Nevertheless, HMM-HMT retain the RR up to approximately 20 training patterns, where begins a weak trend from 35.08% to 30.73%. This important robustness of the proposed model can be attributed to the better capability for modeling the relevant information of features in the wavelet domain.

## 4 Conclusions

The proposed algorithms for HMM-HMT allows learning from variable-length sequences in the wavelet domain. The training algorithms were derived using the EM framework, resulting in a set of learning rules with a simple structure.

The recognition rates obtained for classification were very competitive, even in comparison with the state-of-the-art technologies in this application domain. The proposed post-processing for the DWT feature extraction resulted in very important improvements of the recognition rates. In this empirical tests, the novel architecture and training algorithm demonstrated to be the most robust to the reduction of the amount of training data.

Future works will be oriented to reduce the computational cost of the training algorithms and to test the proposed model in continuous speech recognition and contaminating the speech with non-stationary noises.

*Acknowledgment.* This work is supported by the National Research Council for Science and Technology (CONICET), the National Agency for the Promotion of Science and Technology (ANPCyT-UNL PICT 11-25984 and ANPCyT-UNER PICT 11-12700), and the National University of Litoral (UNL, project CAID 012-72).

## References

1. Crouse, M., Nowak, R., Baraniuk, R.: Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing* 46(4), 886–902 (1998)
2. Mallat, S.: *A Wavelet Tour of signal Processing*, 2nd edn. Academic Press, London (1999)
3. Fan, G., Xia, X.G.: Improved hidden Markov models in the wavelet-domain. *IEEE Transactions on Signal Processing* 49(1), 115–120 (2001)
4. Durand, J.B., Gonçalves, P., Guédon, Y.: Computational methods for hidden Markov trees. *IEEE Transactions on Signal Processing* 52(9), 2551–2560 (2004)
5. Sebe, N., Cohen, I., Garg, A., Huang, T.: *Machine Learning in Computer Vision*. Springer, Heidelberg (2005)
6. Baldi, P., Brunak, S.: *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, Massachusetts (2001)

7. Jelinek, F.: *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, Massachusetts (1999)
8. Kim, S., Smyth, P.: Segmental Hidden Markov Models with Random Effects for Waveform Modeling. *Journal of Machine Learning Research* 7, 945–969 (2006)
9. Bishop, C.: *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford (1995)
10. Rabiner, L., Juang, B.: *Fundamentals of Speech Recognition*. Prentice-Hall, New Jersey (1993)
11. Bengio, Y.: Markovian Models for Sequential Data. *Neural Computing Surveys* 2, 129–162 (1999)
12. Fine, S., Singer, Y., Tishby, N.: The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning* 32(1), 41–62 (1998)
13. Murphy, K., Paskin, M.: Linear time inference in hierarchical HMMs. In: Dietterich, T., Becker, S., Ghahramani, Z. (eds.) *Advances in Neural Information Processing Systems* 14, vol. 14, MIT Press, Cambridge (2002)
14. Willsky, A.: Multiresolution Markov models for signal and image processing. *Proceedings of the IEEE* 90(8), 1396–1458 (2002)
15. Dasgupta, N., Runkle, P., Couchman, L., Carin, L.: Dual hidden Markov model for characterizing wavelet coefficients from multi-aspect scattering data. *Signal Processing* 81(6), 1303–1316 (2001)
16. Lu, J., Carin, L.: HMM-based multiresolution image segmentation. *IEEE International Conference on Acoustics, Speech and Signal Processing* 4, 3357–3360 (2002)
17. Bengio, S., Bourlard, H., Weber, K.: An EM algorithm for HMMs with emission distributions represented by HMMs. Technical Report IDIAP-RR 11, Martigny, Switzerland (2000)
18. Weber, K., Ikbal, S., Bengio, S., Bourlard, H.: Robust speech recognition and feature extraction using HMM2. *Computer Speech & Language* 17(2-3), 195–211 (2003)
19. Bharadwaj, P., Carin, L.: Infrared-image classification using hidden Markov trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(10), 1394–1398 (2002)
20. Ichir, M., Mohammad-Djafari, A.: Hidden Markov models for wavelet-based blind source separation. *IEEE Transactions on Image Processing* 15(7), 1887–1899 (2006)
21. Zue, V., Sneff, S., Glass, J.: Speech database development: TIMIT and beyond. *Speech Communication* 9(4), 351–356 (1990)
22. Stevens, K.: *Acoustic phonetics*. MIT Press, Cambridge (1998)
23. Mallat, S.: A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(7), 674–693 (1989)
24. Daubechies, I.: *Ten Lectures on Wavelets*. In: Number 61 in CBMS-NSF Series in Applied Mathematics, SIAM, Philadelphia (1992)
25. Evangelista, G.: Pitch-synchronous wavelet representations of speech and music signals. *IEEE Transactions on Signal Processing* 41(12), 3313–3330 (1993)
26. Chan, C.P., Ching, P.C., Leea, T.: Noisy speech recognition using de-noised multiresolution analysis acoustic features. *J. Acoust. Soc. Am.* 110(5), 2567–2574 (2001)
27. Farooq, O., Datta, S.: Mel Filter-Like Admissible Wavelet Packet Structure for Speech Recognition. *IEEE Signal Processing Letters* 8(7) (2001)

# Assessment of Personal Importance Based on Social Networks

Przemysław Kazienko and Katarzyna Musiał

Wrocław University of Technology, Institute of Applied Computer Science  
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland  
{kazienko, katarzyna.musial}@pwr.wroc.pl

**Abstract.** People that interact, cooperate or share common activities within information systems can be treated as a social network. The analysis of individual social standings appears to be a crucial element for the assessment of personal importance of each member within such weighted social network. The new measure of person significance – social position that depends on both the strength of relationships an individual maintains and social positions of all their acquaintances, together with its basic features and comparative experiments are presented in this paper.

**Keywords:** personal importance, social position, social network analysis.

## 1 Introduction

The constantly growing popularity of ubiquitous, web-based services, in which people can communicate with one another and share their interests, reveals that human relationships have moved more and more from the real world to the virtual world. This process has begun since the first email service started, but recently we can observe its substantial expansion in many systems such as instant messengers, blogs, wikis, dating systems, multimedia publishing systems, e.g. Flickr or YouTube, and many others. Humans with their spontaneous but also social behavior are the most significant, and simultaneously the less predictable element in each of such systems.

People who interact with one another or share common activities form a social network. Overall, a social network is treated as a finite set of individuals, by sociologists called actors, who are the nodes of that network, and ties that are the relationships between them [9, 10, 18]. In this paper, ties reflect behavioral interactions between human beings or their common activities. For example, a couple of people who send email messages to each other, who learn the same lessons in an e-learning system, talk to each other or who comment on the same internet blogs are in a mutual relation. Thus, the social network is defined in this article as follows: a social network  $SN$  is a tuple  $SN=(M,R)$ , where  $M$  is the finite set of network members, that communicate with one another and  $R$  is the set of social relationships (ties) that join pairs of distinct network members,  $R$  is a subset of  $M \times M$ , i.e.  $R=\{(m_i,m_j): m_i \in M, m_j \in M, i \neq j\}$ ,  $card(M)>1$ . The set of members  $M$  must not contain isolated members. Every network member possesses at least one relationship within the given social network i.e. the graph that represents the social network is connected.

Based on the data derived from a source system, we can build a social network of its members and then analyze the position of each person within the network. This would help to discover individuals who possess the highest social statement and probably the highest level of personal trust and importance. These people can represent the entire community or be encouraged to initiate new kinds of actions while human beings with the lowest social position may be stimulated for greater activity.

## 2 Related Work

Social network analysis provides many measures to determine the characteristic of a person within the social network like centrality, prestige, reachability, connectivity [10, 18]. All of them indicate the personal importance of an individual in the network. The first two: centrality and prestige have been analyzed and compared to the new proposed measure – social position.

Centrality is a measure that is commonly used to identify the most significant members within the network [5]. There are some approaches to evaluate the centrality [8] of a single member of the social network: closeness centrality, betweenness centrality, and degree centrality.

Degree centrality  $DC(x)$  of person  $x$  in network takes into account the outdegree of person  $x$  [8, 16], i.e. the number of individuals that a given member  $x$  is connected to. In this case one's degree centrality is higher when this person communicates with the greater number of people. It is expressed by the normalized number of neighbors that are adjacent from the given person, as follows:

$$DC(x) = o(x) / (n-1), \quad (1)$$

where:  $o(x)$  – the number of the first level neighbors that are adjacent from  $x$ ;  $n$  – the total number of users in the social network.

A slightly different approach presented Bonacich. He claimed that person's degree centrality depends not only on the number of members that one communicates with but also on the number of connections that have one's acquaintances [10].

Both of the degree centrality measures have some shortcomings. They take into consideration only direct relationships of the member or the connections of the one's direct neighborhood and this disadvantage can cause a person is a central actor but only locally [10]. In order to cope with that problem the closeness centrality can be used. This measure pinpoints how close an individual is to all the others within the social network [2]. Its main idea is that the user takes the central position if they can quickly contact other users in the network. The similar idea was studied for hypertext systems [3].

The closeness centrality  $CC(x)$  of user  $x$  tightly depends on the geodesic distance, i.e. the shortest paths from user  $x$  to all other people in the social network [15] and is calculated as follows:

$$CC(x) = (n-1) / \sum_{i=1}^n d(x, y), \quad (2)$$

where  $d(x, y)$  is the length of the shortest path from user  $x$  to  $y$ .

The closeness centrality based on the geodesic distance is the widely used solution but of course not the only one that exists. The other approach to the closeness central-

ity respects the number and additionally the length of all independent paths between a pair of network members [10].

Betweenness centrality of person  $x$  pinpoints to what extend  $x$  is between other network members. It can be calculated only for undirected relationships by dividing the number of shortest geodesic distances (paths) from  $y$  to  $z$  by the number of shortest geodesic distances from  $y$  to  $z$  that pass through user  $x$ . This calculation is repeated for all pairs of individuals  $y$  and  $z$ , excluding  $x$ . Betweenness centrality of person  $x$  is the sum of all the outcomes [7]. Member  $x$  is more important if there are many people in the social network that must communicate with  $x$  in order to make relationships with other network members [10].

The second feature that characterizes an individual in the social network and enables to identify the most powerful members is prestige. Similarly to centrality, prestige can be calculated in various ways, e.g. proximity prestige, rank prestige, and degree prestige. The degree prestige is based on the indegree number so it takes into account the number of users that are adjacent to a particular member of the community [18]. In other words, more prominent people are those who received more nominations from members of the community [1]. The degree prestige  $DP(x)$  of user  $x$  can be described with the following formula:

$$DP(x) = i(x) / (n-1), \tag{3}$$

where  $i(x)$  – number of users from the first level neighborhood that are adjacent to  $x$ .

Proximity prestige  $PP(x)$ , in contrary to closeness centrality, shows how close are all users within the social community to user  $x$  [18]. This measure depends on the geodesic distances of all users to  $x$ :

$$PP(x) = \frac{\frac{k_x}{n-1}}{\frac{1}{k_x} \sum_{i=1}^{k_x} d(y,x)} = \frac{(k_x)^2}{(n-1) \cdot \sum_{i=1}^{k_x} d(y,x)}, \tag{4}$$

where  $k_x$  – the number of users who can reach user  $x$ , i.e. there exist paths from these users to user  $x$ .

Similarly to degree centrality and closeness centrality, there are several approaches to degree prestige and proximity prestige calculation. They correspond to centrality measures described above.

The rank prestige, which is also called status prestige [18], is measured based on the status of users in the network and depends not only on geodesic distance and number of relationships, but also on the status of users connected with the user.

Brin and Page introduced another measure called PageRank and used it to assess the value and importance of web pages [4]. The PageRank value of a web page takes into consideration PageRank of all other pages that link to this particular one. The main difference between PageRank and social position function proposed in this paper is the existence and meaning of commitment function. In PageRank, all links have the same weight and importance whereas social position makes the quantitative distinction between the strengths of individual relationships.

All above measures can be applied not only to identify the most significant persons but also to find the groups of people who are the most powerful ones [18].

### 3 Assessment of Personal Importance Based on Social Position

There is a great need to assess the personal significance within the weighted social network. The new measure called *social position* enables to estimate the importance of the person within the social community. It takes into consideration the significance of the nearest neighbors of a person as well as the quality of one's mutual relationships. The social position refers to the standing and potential social capital of an individual in the network [14]. The personal importance of the community member in the weighted network tightly depends on the strength of the relationships that this member maintains as well as on the social positions of their acquaintances, i.e. the first level neighbors. In other words, the user's social position is inherited from others but the level of inheritance depends on the activity of the members directed to this person, i.e. intensity of common interaction, cooperation or communication. Social position function  $SP(x)$  of individual  $x$  respects both the value of social positions of person  $x$ 's acquaintances as well as their activities in relation to  $x$ :

$$SP(x) = (1 - \varepsilon) + \varepsilon \cdot (SP(y_1) \cdot C(y_1 \rightarrow x) + \dots + SP(y_m) \cdot C(y_m \rightarrow x)), \quad (5)$$

where:  $\varepsilon$  – the constant coefficient from the range  $[0,1]$ ;  $y_1, \dots, y_m$  – acquaintances of  $x$ , i.e. people who are in the direct relation to person  $x$ ;  $m$  – the number of acquaintances of person  $x$ ;  $C(y_1 \rightarrow x), \dots, C(y_m \rightarrow x)$  – the function that denotes the contribution in activity of person  $y_1, \dots, y_m$ , respectively, directed to individual  $x$ .

The contribution in activity  $C(y \rightarrow x)$  can usually be obtained in an automatic way, e.g. simply as the number of emails sent by  $y$  to  $x$  in relation to total  $y$ 's emails or the duration of phone calls made by  $y$  to  $x$  in relation to total duration of  $y$ 's phone calls.

The social position is calculated in the iterative way. It means that the left side hand of Eq. (5) is the result of iteration while the right side hand is the input according to the following formula:

$$SP_{n+1}(x) = (1 - \varepsilon) + \varepsilon \cdot (SP_n(y_1) \cdot C(y_1 \rightarrow x) + \dots + SP_n(y_m) \cdot C(y_m \rightarrow x)), \quad (6)$$

where  $SP_{n+1}(x)$  and  $SP_n(x)$  is the social position of member  $x$  after the  $n+1$ th and  $n$ th iteration, respectively.

To perform the first iteration, we also need to have an initial value of social position  $SP_0(x)$  for all  $x \in M$ :

$$SP_1(x) = (1 - \varepsilon) + \varepsilon \cdot (SP_0(y_1) \cdot C(y_1 \rightarrow x) + \dots + SP_0(y_m) \cdot C(y_m \rightarrow x)). \quad (7)$$

To calculate the social position of the person within the weighted social network the convergent, iterative algorithm is used. This means that we have to specify an appropriate stop condition and the calculation are iteratively repeated until it is reached. There can be different versions of stop condition, e.g. no difference in social position values to the precision of 5 digits after the decimal point for all the users in two following iterations.

There are some prerequisites that have to be taken into account. Firstly, isolated members who do not have any incoming or outgoing relationships in the network are rejected from further calculations. Next, the contribution of the activity  $C(y \rightarrow x_i)$  for all the members  $y$  who are in some relationships with network members  $x_i$  but are not active within these relationships at all, is distributed equally among all user  $y$ 's acquaintances  $x_i$ . The reason being is that the sum of contributions per member equals 1.



Social position function has several specific features. One of them is that the initial values of social positions influence the number of iterations but not their final values. Furthermore, the sum of all the social positions within the network is convergent to the total number of users in the network. The distribution of  $SP$  values depends on  $\varepsilon$ , which also influences the number of iterations. It means that, the greater  $\varepsilon$ , the more iterations must be done. Moreover, the social position of an individual is nearly linearly dependent on the value of  $\varepsilon$ . Additionally, the greater  $\varepsilon$ , the greater the difference between maximum and minimum value of  $SP$  and the greater the standard deviation. Regardless the changes of  $\varepsilon$ , the order of persons within human community based on their social positions remains fixed, although the difference in  $SP$  between two following members may change. Furthermore, the average social position is convergent to 1 and does not depend on the value of  $\varepsilon$ .

Social position is in a sense an indegree measure since it depends on the number of people who communicate to an evaluated community member – compare it to degree prestige  $DP$ , (Eq. 3). The main difference between  $SP$  and  $DP$  is that  $SP$  not only includes the quantity of the neighbors but also their quality, i.e. their social positions. Additionally,  $SP(x)$  makes use of the quality of relationships directed to  $x$ . In other words, the more  $y$  communicates to  $x$ , the more person  $y$  transfers their  $SP(y)$  to  $x$ . If this communication has a major contribution in the entire activities of  $y$  then we can suppose that  $x$  must be an important person for  $y$ . In this way we utilize more precise quantitative data, i.e. weights of relationships and value of acquaintances, unlike binary data as at degree prestige.

Note that the same social position can be achieved by person  $x$  either if  $x$  has many relationships with people who have a medium social position or if  $x$  has only a few relationships but with participants who possess high social position.

## 4 Experiments

The aim of the experiments is to compare the measures of centrality and prestige with the proposed social position. The research has been performed on two social networks: *Thurman office network* and *CAMP92*. To calculate social position some assumptions should be made. Firstly, the initial social positions equal 1 are established for every member in both networks. Secondly, the value of  $\varepsilon$  is 0.9 and the stop condition is: no difference in social position values to the precision of 5 digits after the decimal point for all the users in two following iterations.

### 4.1 Thurman Office Social Network

The Thurman office social network is a non-symmetric network of 15 people who worked in one company (Fig. 1). Thurman spent 16 months observing the interactions among employees in the office of a large corporation [17]. The sociomatrix presented in Table 1 is the transformed, original, non-symmetric matrix of relationships within the Thurman office social network. A non-zero value means that employee  $x$  from the row is in relationship to employee  $y$  from the column, e.g. *President* (10) contacts with *Amy* (4). Note that *Amy* does not communicate to *President*. To evaluate social position  $SP$  (Eq. 5) the contributions in activity for each network member are

established by dividing one (1 was in the original network) by the number of individual's relationships, e.g. *Emma* communicates with nine users so her contribution of activity to every her acquaintance equals  $\frac{1}{9}$ . Other measures have been calculated according to the appropriate formulas from sec. 2: degree centrality (*DC*) (Eq. 1), closeness centrality (*CC*) (Eq. 2), degree prestige (*DP*) (Eq. 3), and proximity prestige (*PP*) (Eq. 4). They have been compared to *SP* in Fig. 2 and Fig. 3.

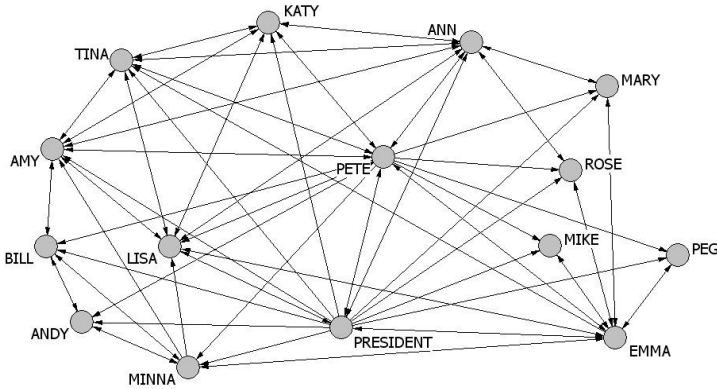


Fig. 1. Graph presentation of Thurman office social network

Table 1. Sociomatrix presentation of Thurman office social network

Individual	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	3
1. Emma			$\frac{1}{9}$		$\frac{1}{9}$	$\frac{1}{9}$		$\frac{1}{9}$		$\frac{1}{9}$		$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
2. Ann			$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$			$\frac{1}{8}$		$\frac{1}{8}$	$\frac{1}{8}$			$\frac{1}{8}$
3. Pete	$\frac{1}{14}$	$\frac{1}{14}$		$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	
4. Amy		$\frac{1}{6}$	$\frac{1}{6}$		$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$		$\frac{1}{6}$							$\frac{1}{6}$
5. Lisa	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$		$\frac{1}{7}$	$\frac{1}{7}$			$\frac{1}{7}$						$\frac{1}{7}$
6. Tina		$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$		$\frac{1}{5}$									$\frac{1}{5}$
7. Katy		$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$										$\frac{1}{5}$
8. Minna	$\frac{1}{5}$			$\frac{1}{5}$	$\frac{1}{5}$				$\frac{1}{5}$		$\frac{1}{5}$					
9. Bill				$\frac{1}{3}$				$\frac{1}{3}$			$\frac{1}{3}$					
10. President	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$
11. Andy			$\frac{1}{3}$					$\frac{1}{3}$	$\frac{1}{3}$							$\frac{1}{3}$
12. Mary	$\frac{1}{2}$	$\frac{1}{2}$														
13. Rose	$\frac{1}{2}$	$\frac{1}{2}$														
14. Mike	1															
15. Peg	1															

Based on the obtained values five separate rankings have been created. The positions of each person in every ranking are presented in Table 2. Note that the order of people in respect to social position and degree centrality or closeness centrality varies

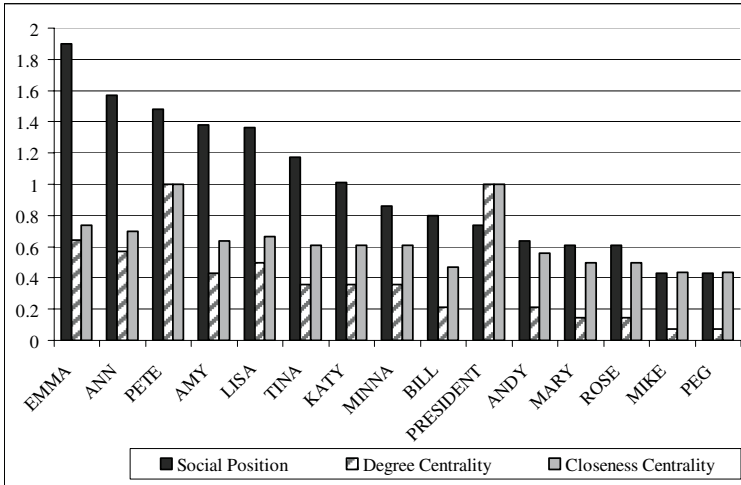


Fig. 2. The comparison of centrality measures to social position in Thurman network

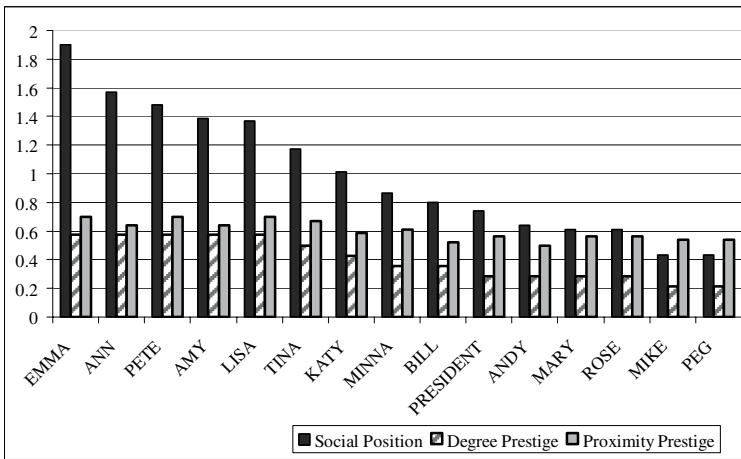


Fig. 3. The comparison of prestige measures to social position in Thurman office network

a lot, i.e. *President* obtains the 10th position in the ranking according to social position whereas *President* holds the first position according to both centrality measures. On the other hand, the rankings of people based on social position and degree prestige are quite similar (Table 2), even though the distribution of user social position is greater. However, social position provides better opportunity to distinguish networks members within the network in opposite to both prestige measures.

**Table 2.** The positions in rankings for the analyzed measures in Thurman office social network

Individual	<i>SP</i>	<i>DC</i>	<i>CC</i>	<i>DP</i>	<i>PP</i>
Emma	1	3	3	1	1
Ann	2	4	4	1	5
Pete	3	1	1	1	1
Amy	4	6	6	1	5
Lisa	5	5	5	1	1
Tina	6	7	7	6	4
Katy	7	7	7	7	8
Minna	8	7	7	8	7
Bill	9	10	13	8	15
President	10	1	1	10	9
Andy	11	10	10	10	12
Mary	12	12	11	10	9
Rose	12	12	11	10	9
Mike	14	14	14	14	13
Peg	14	14	14	14	13

**Table 3.** Sociomatrix presentation of CAMP92 social network

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1 Holly		11	1	17	15	5	10	12	14	6	3	16	4	13	2	7	9	8
2 Brazey	14		10	13	8	9	11	12	3	1	17	7	5	2	4	16	15	6
3 Carol	2	8		17	16	14	15	7	11	1	5	4	13	3	6	12	10	9
4 Pam	14	11	13		12	15	17	16	9	2	5	8	10	7	3	4	6	1
5 Pat	16	12	15	5		17	13	14	4	1	9	3	6	2	8	10	11	7
6 Jennie	9	10	13	15	17		14	16	4	1	6	7	2	3	5	12	11	8
7 Pauline	8	12	15	17	16	13		14	5	3	7	2	10	1	4	9	11	6
8 Ann	13	10	14	16	11	17	15		9	3	4	6	12	8	2	7	5	1
9 Michael	15	3	13	12	11	5	7	8		4	1	17	6	16	14	9	10	2
10 Bill	10	9	4	3	7	1	5	2	17		8	16	6	15	11	13	12	14
11 Lee	8	15	12	5	14	4	3	1	7	2		9	10	11	6	17	16	13
12 Don	16	5	3	13	12	11	10	6	17	1	8		4	15	14	7	9	2
13 John	2	13	14	10	6	4	17	9	8	5	1	7		11	16	12	3	15
14 Harry	16	2	3	8	7	1	4	10	17	13	12	15	14		9	11	6	5
15 Gery	5	12	9	4	7	3	6	2	15	1	10	11	13	8		16	14	17
16 Steve	4	13	12	3	9	7	10	5	6	1	16	11	8	2	14		17	15
17 Bert	13	14	8	11	3	7	10	9	4	5	16	6	1	2	12	17		15
18 Russ	14	10	9	3	5	11	1	2	6	7	13	8	12	4	17	15	16	

The information that *Emma*, *Ann*, *Pete*, *Amy*, and *Lisa* have the same, greatest degree prestige is insignificant since it results from the number of other users who are adjacent to them. The social position measure  $SP(x)$  takes into consideration not only the number of members who communicates to the evaluated person  $x$  but also their

social positions and their contribution of activity directed to  $x$ . These properties mean that *Emma* is the person with the highest social position in the network because *Mike* and *Peg* communicate only with *Emma* so they transfer their entire social positions to her. The prestige measurements do not respect these features and this appears to be important at assessing the importance of an individual within the social network.

## 4.2 CAMP92 Social Network

The second set of experiments performed in the CAMP92 social network, which consists of 18 people and where every user is connected to all the others. This data was collected by Borgatti, *at al.* at the 1992 NSF Summer Institute on Research Methods in Cultural Anthropology by creating a list of all persons and asking each respondent to sort the participants according to the intensity of the interaction they had with each person since the beginning of the course. The value "17" in the ranking indicates the most frequent interaction while a "1" denotes the least interaction.

Similarly to the tests on Thurman office social network, five different measures have been calculated (Fig. 4 and Fig. 5). In order to evaluate the contributions in activity, necessary for  $SP$ , each value in Table 3 has been normalized with the sum of its row. Based on these values five rankings have been created (Table 4).

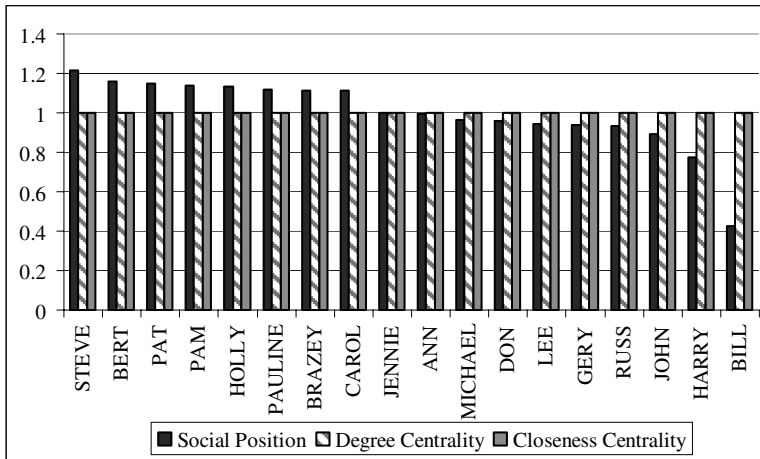


Fig. 4. The comparison of centrality measures to social position for CAMP92 network

As we can see, values of all centrality and prestige measures equal one for all users. This reveals one of the main shortcomings of these measures. If the social network is a clique [6], where every person is connected to all others, then centrality and prestige measures become useless because according to these measures all network members are equally important. In consequence, the social position turns out to be much more adequate and meaningful. The highest social position is possessed by

Steve since other people interact with him more often and more willingly than with others in the network. On the other hand, Bill has the lowest social position because others in the network communicate with him the least (Table 3).

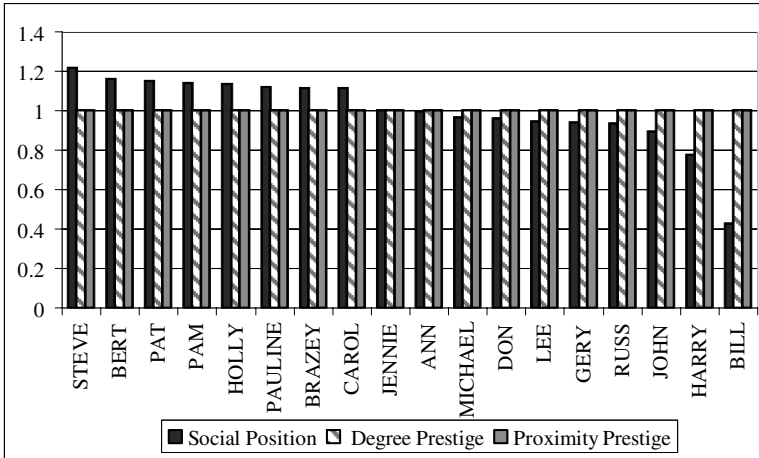


Fig. 5. The comparison of prestige measures to social position for CAMP92 network

Table 4. Positions in rankings based on five different user measures for CAMP92 network

Individual	SP	DC	CC	DP	PP
Steve	1	1	1	1	1
Bert	2	1	1	1	1
Pat	3	1	1	1	1
Holly	4	1	1	1	1
Pam	5	1	1	1	1
Pauline	6	1	1	1	1
Brazey	7	1	1	1	1
Carol	8	1	1	1	1
Jennie	9	1	1	1	1
Ann	10	1	1	1	1
Michael	11	1	1	1	1
Don	12	1	1	1	1
Lee	13	1	1	1	1
Gery	14	1	1	1	1
Russ	15	1	1	1	1
John	16	1	1	1	1
Harry	17	1	1	1	1
Bill	18	1	1	1	1

### 5 Conclusions and Future Work

A social network can be created utilizing data available in many multi-user IT systems where people communicate or cooperate with one another. Thus, human communities exist almost everywhere.

Social position presented in the paper is a wide-ranging measure for the assessment of personal importance within the weighted social network. It not only includes information about the number of people who are in relationship with the evaluated individual, but it also respects their social standing and quality of mutual relationship. According to the experiments conducted, social position appears to be better than other measures used in the social network analysis for assessing the importance of the person.

Future work will focus on the application of social position in different domains like recommendation systems [13], targeted advertising or marketing [12, 19], and telecommunication [11].

**Acknowledgments.** This work was partly supported by The Polish Ministry of Science and Higher Education, grant no. N516 037 31/3708.

## References

1. Alexander, C.N.: A method for processing sociometric data. *Sociometry* 26, 268–269 (1963)
2. Bavelas, A.: Communication patterns in task – oriented groups. *Journal of the Acoustical Society of America* 22, 271–282 (1950)
3. Botafogo, R.A., Rivlin, E., Shneiderman, B.: Structural analysis of hypertexts: identifying hierarchies and useful metrics. *ACM Trans. on Information Systems* 10(2), 142–180 (1992)
4. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. WWW7, 1998, also *Computer Networks and ISDN Systems* 30(1-7), 107–117 (1998)
5. Everett, M., Borgatti, S.: Extending Centrality. In: *Models and Methods in Social Network Analysis*, Cambridge University Press, New York (2005)
6. Falzon, L.: Determining groups from the clique structure in large social networks. *Social Networks* 22, 159–172 (2000)
7. Freeman, L.C.: A set of measures of centrality based on betweenness. *Sociometry* 40, 35–41 (1977)
8. Freeman, L.C.: Centrality in social networks Conceptual clarification. *Social Networks* 1(3), 215–239 (1979)
9. Garton, L., Haythornthwaite, C., Wellman, B.: Studying Online Social Networks. *Journal of Computer-Mediated Communication* 3(1) (1997)
10. Hanneman, R., Riddle, M.: Introduction to social network methods. In: Online textbook, 01.04.2006 (2006), available at <http://faculty.ucr.edu/hanneman/nettext/>
11. Kazienko, P.: Expansion of Telecommunication Social Networks. In: Luo, Y. (ed.) *CDVE 2007*. LNCS, vol. 4674, pp. 404–412. Springer, Heidelberg (2007)
12. Kazienko, P., Adamski, M.: AdROSA - Adaptive Personalization of Web Advertising. *Information Sciences* 177(11), 2269–2295 (2007)
13. Kazienko, P., Musiał, K.: Recommendation Framework for Online Social Networks. In: *AWIC 2006*. *Studies in Computational Intelligence*, vol. 23, pp. 111–120. Springer, Heidelberg (2006)
14. Kazienko, P., Musiał, K.: Social Capital in Online Social Networks. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) *KES 2006*. LNCS (LNAI), vol. 4252, pp. 417–424. Springer, Heidelberg (2006)
15. Sabidussi, G.: The centrality index of a graph. *Psychometrica*, 31 (1966)
16. Shaw, M.E.: Group structure and the behavior of individuals in small groups. *Journal of Psychology* 38, 139–149 (1954)
17. Thurman, B.: In the office: Networks and coalitions. *Social Networks* 2, 47–63 (1979)
18. Wasserman, S., Faust, K.: *Social network analysis: Methods and applications*. Cambridge University Press, New York (1994)
19. Yang, W.S., Dia, J.B., Cheng, H.Ch., Lin, H.T.: Mining Social Networks for Targeted Advertising. In: *HICSS-39 2006*, IEEE Computer Society Press, Los Alamitos (2006)

# Optimization Procedure for Predicting Nonlinear Time Series Based on a Non-Gaussian Noise Model

Frank Emmert-Streib<sup>1,\*</sup> and Matthias Dehmer<sup>2</sup>

<sup>1</sup> Stowers Institute for Medical Research  
1000 E. 50th Street, Kansas City, MO 64110, USA  
`fes99@u.washington.edu`

<sup>2</sup> Discrete Mathematics and Geometry  
Vienna University of Technology  
Wiedner Hauptstrasse 8-10, A-1040 Vienna, Austria  
`mdehmer@geometrie.tuwien.ac.at`

**Abstract.** In this article we investigate the influence of a Pareto-like noise model on the performance of an artificial neural network used to predict a nonlinear time series. A Pareto-like noise model is, in contrast to a Gaussian noise model, based on a power law distribution which has long tails compared to a Gaussian distribution. This allows for larger fluctuations in the deviation between predicted and observed values of the time series. We define an optimization procedure that minimizes the mean squared error of the predicted time series by maximizing the likelihood function based on the Pareto-like noise model. Numerical results for an artificial time series show that this noise model gives better results than a model based on Gaussian noise demonstrating that by allowing larger fluctuations the parameter space of the likelihood function can be search more efficiently. As a consequence, our results may indicate a more generic characteristics of optimization problems not restricted to problems from time series prediction.

## 1 Introduction

In machine learning and statistics models based on or utilizing a Gaussian distribution are omnipresent. For example, in nonlinear time series prediction a Gaussian noise model is frequently used to find the maximum likelihood parameters of the model, i.e., the weights between neurons for a neural network model [16]. A formal reason supporting this approach is that one can analytically show that the maximization of the likelihood function for a Gaussian noise model is equivalent to the minimization of the *mean squared error* [2]. Interestingly, recent studies indicate that in nature a different type of distribution is found omnipresently namely power law distributions. For example the magnitude of

---

\* Present address: University of Washington, 1705 NE Pacific St, Box 355065, Seattle WA 98195-5065, USA.



earthquakes [9], word frequency in books [19], the fluctuation of price changes of the stock market [13] or the population size of cities follow a power law to name just a few. For more examples for the occurrences of power laws the reader is referred to [14]. For the following discussion we just want to mention that the two names POWER LAW DISTRIBUTION and PARETO DISTRIBUTION are used exchangeably<sup>1</sup>.

Despite the abundant occurrence of power law distributions in nature its application in, e.g., optimization, prediction or control problems, is surprisingly small. One exception is the *extremal optimization* principle introduced by BÖTTCHER et al. [4]. The key idea of this optimization method is that a power law distribution is used to select local moves in the configuration space. This is in contrast, e.g., to simulated annealing, that utilizes an exponential distribution and, hence, discriminates the selection of configurations with an, abstractly speaking, low fitness. Naively, this is appealing, however, contradicts recent results in statistical physics of driven systems far from equilibrium that exhibit so called *self-organized critical* (SOC) behavior [10] and the omnipresence of power law distributions in such systems. The conclusion one can draw from these studies and BÖTTCHER’s extremal optimization principle is that power law distributions might also be usefully be applicable to problems from nonlinear time series prediction because for difficult (nonlinear) time series the distribution of predicted minus the true value could be more closely to a power law distribution rather than a Gaussian distribution. The crucial point here it that one would like this distribution to be Gaussian because a power law distribution implies the presence of outliers, however, this might not be possible and, hence, a wrong choice of an error model might give worse results.

In this paper we focus on the problem to predict a nonlinear time series [18,3,12,8]. We use an artificial neural network with three layers as predictor and obtain the optimal weights by maximizing a likelihood function for which we assume a Pareto-like noise model, defined in Section 3. We define an optimization procedure that minimizes the mean squared error by maximizing the likelihood function for a Pareto-like noise model. This is not straight forward as in the case for a Gaussian noise model but needs to be done carefully. Further we demonstrate in section 5 numerically for an artificial time series which shows chaotic behavior that a Pareto-like noise model gives better results than a Gaussian noise model used to train the same model based on the maximum likelihood principle. This article finishes in section 6 with conclusions.

## 2 Time Series Prediction

In this section we introduce the necessary notation and the mathematical model we will study. In time series prediction the major objective is to estimate or learn the parameters  $\Theta$  of a predictor  $f_\Theta$  from a given time series  $\{x_t\}_1^T$  of length  $T$  to optimize a given error function  $E$ . A frequently chosen error function to compare

---

<sup>1</sup> In physics it is common to use the term power law distribution whereas in statistics the term Pareto distribution is predominating [14].

the predicted  $f_{\Theta}(x_t) = \tilde{x}_t$  with the observed value  $x_t$  is the *mean squared error* (MSE)

$$E = \frac{1}{T} \sum_{t=1}^T (x_t - \tilde{x}_t)^2 \quad (1)$$

A systematic way to estimate the parameters  $\Theta$  of the predictor  $f_{\Theta}$  is provided by the maximum likelihood method assuming a Gaussian noise model

$$p_{\Theta}(\tilde{x}_t|x_t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_t - \tilde{x}_t)^2}{2\sigma^2}\right) \quad (2)$$

and independence between succeeding observation/prediction points the likelihood function can be written as

$$L(x_t, \tilde{x}_t) = \prod_{t=1}^T p_{\Theta}(x_t, \tilde{x}_t) = \prod_{t=1}^T p_{\Theta}(\tilde{x}_t|x_t)p(x_t) \quad (3)$$

The best predictor is obtained for  $\Theta^*$  which is the maximum of the likelihood function

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}}(L(x_t, \tilde{x}_t)) \quad (4)$$

The reason why the parameter  $\Theta^*$  that maximizes the likelihood corresponds to the parameter that minimizes the best predictor can be seen easily by a short calculation. Taking the negative logarithm of both sides of Eq. 3 results in

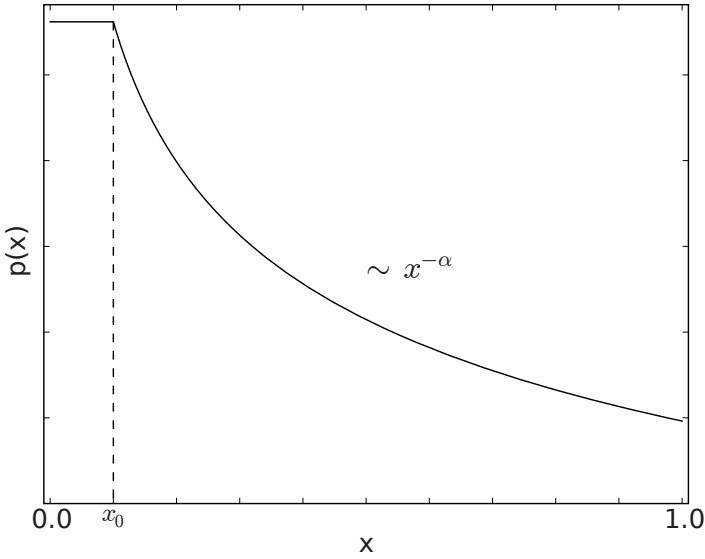
$$-l(x_t, \tilde{x}_t) = -\log(L(x_t, \tilde{x}_t)) = -\sum_{t=1}^T \log(p_{\Theta}(\tilde{x}_t|x_t)) - \sum_{t=1}^T \log(p(x_t)) \quad (5)$$

The last part on the right hand side can be neglected because it is independent of  $\Theta$  and, hence, has no influence on the optimization of the likelihood function. For the noise model in Eq. 2 this gives the mean squared error

$$-l(x_t, \tilde{x}_t) = \frac{1}{2\sigma^2} \sum_{t=1}^T (x_t - \tilde{x}_t)^2 + \frac{T}{2} \log(2\pi\sigma^2) \quad (6)$$

up to constants. This correspondence between a Gaussian noise model and the mean squared error explains its popularity in the context of time series prediction because maximizing the likelihood Eq. 3 assuming a Gaussian noise model minimizes indirectly the mean squared error used to evaluate the predictor [2]. Another reason why a Gaussian noise model is frequently used is provided by the dominating role the Gaussian distribution plays in statistics in general. Despite these arguments, recently, there is increasingly evidence found that power law or Pareto distributions [10, 14] are playing a major role in the description of natural phenomena [10, 14]. This will be discussed in the next section.

<sup>2</sup> In the physical literature the term power law is predominantly used and corresponds to the Pareto distribution in statistics. In the following we will use both terms.



**Fig. 1.** Pareto-like distribution defined in the interval  $[0, 1]$

### 3 Introducing a New Noise Model

In this section we will first define a Pareto-like distribution and then we define a noise model that utilizes such a distribution.

In Fig. 1 we show a Pareto-like distribution. In contrast to a Pareto distribution, the Pareto-like distribution is piecewise defined to include zero in its range of definition. Formally, the distribution is defined as follows.

**Definition 1.** For every positive, real values  $\alpha$  we define the Pareto-like distributions by

$$p(x) = \begin{cases} \frac{1}{C} & : 0 \leq x \leq x_0 \\ \frac{x^{-\alpha}}{C} & : x_0 < x \leq 1 \end{cases} \tag{7}$$

Here, the normalization constant  $C$  is given by

$$C = \int_0^1 p(x)dx = x_0 + \frac{1}{1-\alpha}(1 - x_0^{-\alpha+1}) \tag{8}$$

We want to emphasize, that it is important to extend the range of definition to zero because we want to use the Pareto-like distribution as noise model. That means,  $x$  will be  $x = x_t - \tilde{x}_t$  and the case  $x_t = \tilde{x}_t$  is not only allowed but desired. The likelihood function 3 consists now of two different terms,  $-\log(C)$  if  $0 \leq x \leq x_0$  and  $-\alpha \log(x) + \alpha \log(C)$  if  $x_0 < x \leq 1$ . That means, the Pareto-like distribution as noise model does no longer lead directly or straight forward

to the mean squared error as the Gaussian noise model as shown in Eq. 6. However, in the following we will prove that it is possible to define a procedure, that maximizes the likelihood function with a Pareto-like noise model and leads to the minimization in the squared error.

First, we will provide the following theorem.

**Theorem 1.** *If  $L(\theta)$  and  $L'(\theta')$  are likelihood functions for a Pareto-like noise model,  $x_i, x'_i \in (0, 1]$  and a time series of length  $T$ . Then*

$$L(\theta) \leq L'(\theta') \implies \prod_{i=1}^T x'_i \leq \prod_{i=1}^T x_i \tag{9}$$

Here we used the abbreviation  $x_i = (x_i - \tilde{x}_i)$  (true minus observed value of the time series). For a proof we refer the reader to [6].

From the definition of the mean squared error in Eq. 10 follows immediately

$$E'(\theta') \leq E(\theta) \iff \sum_{i=1}^T x'^2_i \leq \sum_{i=1}^T x^2_i \tag{10}$$

The problem is that

$$L(\theta) \leq L'(\theta') \implies E'(\theta') \leq E(\theta) \tag{11}$$

or analogously

$$\prod_{i=1}^T x'_i \leq \prod_{i=1}^T x_i \implies \sum_{i=1}^T x'^2_i \leq \sum_{i=1}^T x^2_i \tag{12}$$

does not hold in general. That means it is not possible to conclude directly from the maximization of the likelihood function to the minimization of the mean squared error. This can be seen by a counter example for  $T = 2$ . Suppose  $x_1 = 0.95, x_2 = 0.1$  and  $x'_1 = 0.8, x'_2 = 0.7$ . This gives  $L = 0.09, L' = 0.56, E = 0.82, E' = 1.13$  and the inequalities  $L < L'$  and  $E < E'$ . That means, despite the fact that the likelihood  $L'$  is larger than  $L$  the mean squared error  $E'$  is larger than  $E$  and not smaller as it should be. This fact, that a Pareto-like noise model does not optimize the mean squared error but a cost function based on the  $L_1$  norm (and not  $L_2$  as the mean squared error) was already recognized and discussed in [15]. The fact, however, that this does not imply that it is not possible to utilize a Pareto-like noise model to minimize the mean squared error systematically is unknown so far and will be discussed below.

At the first view this looks devastating because we are aiming to minimize the prediction error which is the mean squared error. Interestingly, numerical studies reveal that

$$\text{Prob}(L(\theta) \leq L'(\theta') \implies E'(\theta') \leq E(\theta)) = \alpha \approx 0.77 \tag{13}$$

holds independent of  $T$  if all  $x_i$  and  $x'_i$  are drawn iid from an equal distribution. In the following we utilize this effect in the following way. Suppose we optimize

the likelihood iteratively obtaining the following series of likelihoods  $L_i$  for which holds

$$L_1 < L_2 < L_3 < \dots < L_n \tag{14}$$

A resulting series of errors  $E_i$  obtained by  $E_i = E(f_{\Theta(L_i)})$  would certainly not result in a monotonous series because of Eq. 13 and, hence, would not be convergent. However, we can define a series  $E'_i$  by

$$E'_{i+1} = \begin{cases} E_{i+1} & : E_{i+1} \leq E_i \\ E_i & : \text{else} \end{cases} \tag{15}$$

This new series is apparently monotonous

$$E'_1 \geq E'_2 \geq E'_3 \geq \dots \geq E'_n \tag{16}$$

and, hence, convergent. It hold

$$\lim_{i \rightarrow \infty} E'_i = E \tag{17}$$

That means this procedure ensures the maximization of  $L$  results in a systematic minimization of  $E'$ .

---

**Algorithm 1.** Optimization procedure for the minimization of E

---

- 1: initialize  $\Theta_0$
  - 2:  $L_0 = L(\Theta_0)$
  - 3:  $E_0 = E(f_{\Theta_0(L_0)})$
  - 4:  $E'_0 = E_0$
  - 5: **repeat**
  - 6:   obtain  $\Theta_i$  (e.g. from simulated annealing)
  - 7:    $L_i = L(\Theta_i)$
  - 8:    $E_i = E(f_{\Theta_i(L_i)})$
  - 9:   **if**  $E_{i+1} \leq E_i$  **then**
  - 10:      $E'_{i+1} = E_{i+1}$
  - 11:   **else**
  - 12:      $E'_{i+1} = E_i$
  - 13:   **end if**
  - 14: **until** stop criteria holds
- 

We want to mention that the above procedure 1 works for every  $\alpha$  from Eq. 13 if  $\alpha > 0$  regardless of its actual value. However, a conclusion from the value of  $\alpha$  to the performance of the predictor regarding a certain time series is not possible. For this reason we provide in section 5 numerical results.

## 4 Predicting Time Series with ANN

So far we talked only in an abstract way about the predictor  $f_{\Theta}$ . In this section we will define it explicitly. We use a feedforward neural network with three layers.

Because the major topic of this paper is not to discuss methods optimizing the structure of a neural network but the noise model used to optimize the prediction we choose a network structure appropriate for our problem and keep it fixed. We use a 5 – 10 – 1 structure -  $I = 5$  input,  $H = 10$  hidden and  $O = 1$  output neuron. One output neuron is enough because we want to predict only one step ahead of the time series. The  $H$  hidden neurons are fully connected to the  $I$  input neurons. The activity of the hidden neurons is given by

$$x_i^H = T\left(\sum_j^H w_{ij}^I x_j^I\right) \quad (18)$$

with the sigmoid transfer function

$$T(x) = (1 + \exp(-ax))^{-1} \quad (19)$$

The parameter,  $a$ , of the sigmoid can be set to one without loss of generality [18]. The activity of the output neuron is just the linear, weighted sum of the activity of the hidden neurons,

$$x^O = \begin{cases} \sum_i w_i^H x_i^H & : 0 \leq \sum_i w_i^H x_i^H \leq 1 \\ 0 & : else \end{cases} \quad (20)$$

In the following we will assume that the time series can assume values in  $[0, 1]$  which can always be obtained by proper normalization. For this reason, we allow the output neuron only to assume values in this range. The hidden and output neurons have a bias which is adjustable by introducing an additional weight in the form

$$x^O = \sum_i w_i^H x_i^H + w_b^O b^O \quad (21)$$

for the output neuron. We set  $b^O = 1$  and allow  $w_b^O$  to be adapted. Similarly, we introduce a bias for the hidden neurons. The weights  $w^H$  and  $w^O$  are real valued. We use a three layer neural network, because it has been shown, that a feedforward neural network with one hidden layer and sigmoidal transfer function, is capable to approximate every continuous real valued function [5, 7].

The parameters (weights) of the neural network are found by maximizing the likelihood function for a given noise model. This maximization is done numerically by application of simulated annealing [11], a stochastic optimization method.

## 5 Numerical Results

In this section we demonstrate the applicability of a Pareto-like noise model for nonlinear time series. We use the logistic map in the chaotic regime [17] because it is known that chaotic time series are difficult to predict.

### 5.1 Logistic Map

We generate an artificial time series with the logistic map [17]

$$x_{t+1} = rx_t(1 - x_t) \tag{22}$$

and  $r = 4.0$  leading to chaotic time series indicated by a positive Lyapunov exponent (in our case it is  $\sim 1.0$ ). We investigated the influence of  $\alpha$  and  $x_0$ , the parameters of the Pareto-like distribution, on the performance of the prediction by training the neural network with a time series of length  $T$  for various parameter configurations of  $\alpha$  and  $x_0$  (results not shown). Then we used these parameters to investigate the influence of the length  $T$  of the training set on the performance and compared these results with a Gaussian noise model. A summary of these results is shown in table [1].

First, one can clearly see that learning takes place because the mean squared error is for all configurations for  $T = 100$  and higher below 10%. Second, for all  $T$  we can find parameters of the Pareto-like noise model that give better results than a Gaussian noise model. Interestingly, even the worst parameters give results that are at least acceptable. Third, the advantage using a Pareto-like noise model over a Gaussian noise model becomes larger the shorter the time series  $T$  used to train the predictor. The largest gain is obtained for  $T = 30$ . In this case the improvement (reduction of the mean squared error) is 45% compared to a Gaussian noise model. This means, the more difficult it is to learn the parameters of the predictor - the shorter the time series of the training set is - the better is a Pareto-like noise model. This makes intuitively sense, because the more difficult the problem the more unlikely the distribution of predicted minus observed (true) value of the time series follows a Gaussian distribution. This is a hint that a Pareto-like noise model might be especially beneficial for more difficult problems. For easier problems - longer time series of the training set - both noise models give comparable results and we would expect them for very long training time series to coincide.

**Table 1.** Mean squared error for the Pareto-like and the Gaussian noise model for the logistic map. The number in brackets is the standard deviation.

MSE (STD)	T=30	T=50	T=100	T=200
Gauss	0.086 (0.020)	0.040 (0.010)	0.012 (0.002)	0.008 (0.002)
PL(0.8, 0.10)	0.056 (0.039)	0.134 (0.074)	0.018 (0.012)	0.007 (0.002)
PL(0.8, 0.15)	0.087 (0.050)	0.037 (0.027)	0.011 (0.005)	0.011 (0.006)
PL(1.2, 0.10)	0.125 (0.092)	0.154 (0.017)	0.069 (0.077)	0.017 (0.014)
PL(1.2, 0.15)	0.047 (0.025)	0.033 (0.015)	0.010 (0.004)	0.009 (0.003)

## 6 Conclusions

In this article we introduced an optimization procedure that minimizes the mean squared error by maximizing the likelihood function of a Pareto-like noise model.

The minimization of the mean squared error could not be obtained straight forward from the likelihood function, as for Gaussian noise, but was obtained by introducing a convergent auxiliary series ensuring a monotonous optimization of the mean squared error. We presented numerical results for the prediction of a chaotic time series and investigated the performance of an artificial neural network with three layers in dependence on different parameter configurations. We found that systems parameters can be found resulting in a smaller mean squared error compared to a Gaussian noise model used instead of a Pareto-like noise model. To our knowledge, we are the first using a Pareto-like noise model in this context.

Our approach was motivated by the omnipresence of power law distributions in nature [9,14,19], by recent findings in statistical physics of driven systems far from equilibrium [1,10] and the extremal optimization principle introduced by BÖTTCHER et al. [4]. We think our approach could be of general interest also beyond applications in time series prediction because it suggests the usage of Pareto-like distributions where classically Gaussian distributions are applied. By no means we want to claim that this will always improve the performance of a method but we suggest to investigate the influence of a Pareto-like distribution thoroughly instead of rejecting this distribution in advance. Especially, problems from optimization or related problems where a function needs to be optimized, as in our case the likelihood function, could greatly benefit from the usage of Pareto-like distributions if the underlying problem is hard to solve. Because for such problems it might be desirable to allow jumps in the configuration space that are still centralized around a neighborhood of the current position but also allow further remote jumps with a reasonable probability. Further analysis is needed to study this situation with respect to its generic character regarding problems from machine learning and statistics. On a practical note, it would be interesting to apply our method to data from the stock market or the Dow Jones Index because these real world data are among the most challenging. We hope our work can contribute stimulating such endeavors.

**Acknowledgments.** We would like to thank Dongxiao Zhu and two anonymous reviewers for fruitful discussions. Matthias Dehmer has been supported by the European FP6-NEST-Adventure Programme, contract n° 028875.

## References

1. Bak, P., Tang, T., Wiesenfeld, K.: Self-organized criticality: An explanation of the  $1/f$  noise. *Phys. Rev. Lett.* 59, 381–384 (1987)
2. Baldi, P., Brunck, S.: *Bioinformatics: The machine learning approach*. MIT Press, Cambridge (2001)
3. Baragona, R., Battaglia, F., Cucina, D.: Fitting piecewise linear threshold autoregressive models by means of genetic algorithms. *Computational Statistics and Data Analysis* 47(2), 277–295 (2004)
4. Boettcher, S., Percus, A.: Nature’s way of optimizing. *Artificial Intelligence* 119, 275–286 (2000)



5. Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Math. Contr. Sign. Syst.* 2, 303 (1989)
6. Emmert-Streib, F., Dehmer, M.: Nonlinear Time Series Prediction based on a Power-Law Noise Model. *International Journal of Modern Physics C*, (accepted, 2007)
7. Funahashi, K.: On the approximate realization fo continous mappings by neural networks. *Neural Networks* 2, 183–192 (1989)
8. Giordano, F., La Rocca, M., Perna, C.: Forecasting nonlinear time series with neural network sieve bootstrap. *Computational Statistics and Data Analysis*, (in press, 2006)
9. Gutenberg, B., Richter, R.F.: Frequency of earthquakes in california. *Bulletin of the Seismological Society of America* 34, 185–188 (1944)
10. Jensen, H.J.: *Self-Organized Criticality: Emergent Complex Behavior in Physical and Biological Systems*. Cambridge University Press, Cambridge (1998)
11. Kirkpatrick, S., Gellatt, C., Vecchi, M.: Optimization by simulated annealing. *Science* 220, 671–680 (1983)
12. Liang, F.: Bayesian neural networks for nonlinear time series forecasting. *Statistics and Computing* 15, 13–29 (2005)
13. Mandelbrot, B.B.: The variation of certain speculative prices. *J. Business* 36, 394–419 (1963)
14. Newman, M.E.J.: Power laws, pareto distributions and zipf’s law. *Contemporary Physics* 46, 323–351 (2005)
15. Pontil, M., Mukherjee, S., Girosi, F.: On the noise model of support vector machine regression. Technical report, Center for Biological and Computational Learning and the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology (1998)
16. Roweis, S., Ghahramani, Z.: A unified review of linear gaussian models. *Neural Computation* 11(2), 305–345 (1999)
17. Schuster, H.G.: *Deterministic Chaos*. Wiley VCH Publisher, Chichester (1988)
18. Weigend, A., Huberman, B.A., Rumelhart, D.R.: Predicting the future: A connectionist approach. *International Journal of Neural Systems* 1(3), 193–209 (1990)
19. Zipf, G.K.: *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, Reading, MA (1949)

# An Improved Training Algorithm of Neural Networks for Time Series Forecasting\*

Daiping Hu, Ruiming Wu, Dezhi Chen, and Huiming Dou

Antai College of Economics & Management, Shanghai Jiaotong University,  
200052 Shanghai, China  
{dphu, dwurm, mcc, douhuiming}@sjtu.edu.cn

**Abstract.** Neural network approaches for time series forecasting, which have the property of simpleness, nonlinearity and effectiveness, have been broadly utilized in many domains. In this paper, an improved training algorithm of back-propagation neural network for time series forecasting by using dynamic learning rate in the training process is proposed. The results of some studied cases demonstrate this algorithm can increase the efficiency of neural network training and the precision of forecasts.

## 1 Introduction

Time series forecasting plays a very important role in the domains such as stock market, traffic control, enterprise sales and marketing where a great of data is observed in the form of time series. Varied time series forecasting approaches such as moving average, exponential smooth, Box-Jenkins (Autoregressive integration moving average, ARIMA) have been developed and used widely in real forecasting cases.

Early in 1987, Lapedes [1] recommended that the neural network can be applied in the time series forecasting, and then many researchers focused their study on neural network time series forecasting methods and applications [2, 3, 4, 5, 6]. A lot of neural network forecasting models and related improving strategies have been studied over the passed decades. For their powerful approach ability of arbitrary function, the property of simpleness and effectiveness, neural network approaches for time series forecasting have been broadly utilized in many domains.

Back-propagation neural network (BPNN) is a classical neural network which can be used in single variable or multi-variable time series forecasting. BPNNs are much more simple and easier than ARIMAs in the multi-variable time series modeling and forecasting. However, in the actual forecasting applications, we find two weaknesses of forecasting with BPNNs:

- (1) Network training is time-consuming when processing a great of data samples.
- (2) Precision of forecasts can not meet the high level of satisfaction even though the neural network fitted the original data samples well without over-fitted.

---

\* Supported by National Natural Science Foundation of China (No.70671067) and Youth Research Fund of Antai College of Economics & Management.

To overcome those weaknesses mentioned above, in this paper, we proposed an improved training algorithm of back-propagation neural networks for time series forecasting by using dynamic learning rate in network training process. Some studied cases demonstrated that the improved algorithm can increase the efficiency of neural network training and the precision of forecasts.

This paper is organized as follows: In section 2, we introduce the BPNN approaches for time series forecasting and the regular BPNN training algorithm. In section 3, we describe the improved algorithm. In section 4, we present some studied cases of utilizing the improved algorithm. In section 5, we get conclusions.

## 2 BPNN Approaches for Time Series Forecasting

A multi-variable time series with P variables and some time-point values is denoted as  $(X1_1, X2_1, \dots, Xp_1), (X1_2, X2_2, \dots, Xp_2), \dots, (X1_t, X2_t, \dots, Xp_t), \dots$ . The base principle of forecasting is that we considered there exists some a function relation  $F(\bullet)$  between the forecasting values for K future time-point and the actual values of M previous time-point. The formula is written as:

$$\begin{aligned}
 & ((X1_{t+1}, X2_{t+1}, \dots, Xp_{t+1}), (X1_{t+2}, X2_{t+2}, \dots, Xp_{t+2}), \dots, (X1_{t+K}, X2_{t+K}, \dots, Xp_{t+K})) \\
 & = F((X1_t, X2_t, \dots, Xp_t), (X1_{t-1}, X2_{t-1}, \dots, Xp_{t-1}), \dots, (X1_{t-M+1}, X2_{t-M+1}, \dots, Xp_{t-M+1})) \tag{1}
 \end{aligned}$$

The BPNN consists of an input layer, an output layer and one or more hidden layers. Generally, BPNNs with one hidden layer can be competent for most application problems. Each layer consists of multiple neurons that are connected to neurons in adjacent layers. Since these networks contain many interacting nonlinear neurons in multiple layers, the networks can capture complex relation  $F(\cdot)$  for time series forecasting through network training by using the historical data to get model parameters (connection weights and unit biases)[7, 8]. Parameters are being adjusted to minimize the forecast errors in the iterative network training.

### 2.1 BPNN Types of Time Series Forecasting

Time series forecasting BPNNs (See Fig.1) can be built as two types. One type is single-step forecasting BPNN when  $K=1$ , the other type is multi-step BPNN forecasting when  $K>1$ . The single-step forecasting network has P output units, which are related to P variables and one time can calculate the single-step forecasts for each variable. The forecasts can be used as inputs to make the next step forecasts, known as iterative forecasting. The Multi-step forecasting network has  $K \times P$  output units. One time calculation can make K forecasts for each of all the P variables. It also can make iterative forecasts for far more steps. In actual application of forecasting, single-step forecasting BPNNs are built more often, and then use them to make iterative forecasts.

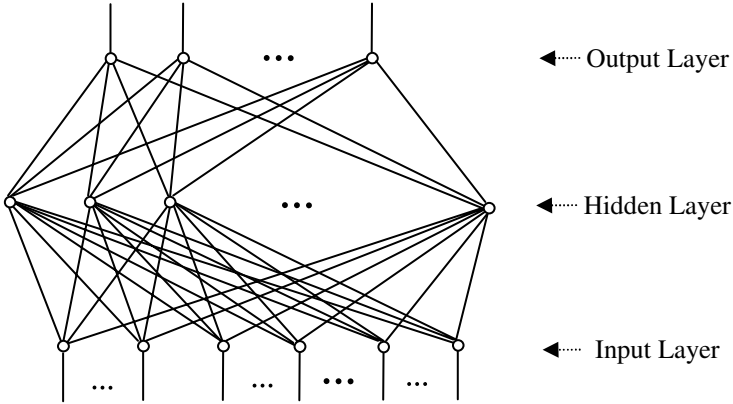


Fig. 1. Multi-step BPNN for time series forecasting

**2.2 Training Algorithm for Time Series Forecasting BPNN**

A single-step forecasting BPNN with three layers is used to describe the training algorithm for time series forecasting. This network has  $Q=P \times M$  input units,  $H$  hidden units and  $P$  output units. For a time series with  $N$  time-point values, it has  $R=N-M$  the training samples, which can be denoted as:

$$S_i = ((x_{i1}, x_{i2}, \dots, x_{iQ}), (y_{i1}, y_{i2}, \dots, y_{iP})) \quad (i=1, 2, \dots, R) \tag{2}$$

where the inputs for the network are:

$$(x_{i1}, x_{i2}, \dots, x_{iQ}) = (X1_i, X2_i, \dots, Xp_i, X1_{i+1}, X2_{i+1}, \dots, Xp_{i+1}, \dots, X1_{i+M-1}, X2_{i+M-1}, \dots, Xp_{i+M-1}) \tag{3}$$

and the network required outputs are:

$$(y_{i1}, y_{i2}, \dots, y_{iP}) = (X1_{i+M}, X2_{i+M}, \dots, Xp_{i+M}) \tag{4}$$

The network training launches with randomly created initial values of weights and biases for all network units. The network training algorithm is an iterative process. The calculation method for each training iteration can be described as follows:

- (1) Calculate the hidden layer unit outputs:

$$\hat{z}_{ij} = S\left(\sum_{k=1}^Q wh_{jk} x_{ik} + bh_j\right) \quad (j=1, 2, \dots, H) \tag{5}$$

where

$$S(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

$wh$  represents weight,  $bh$  represents bias.

(2) Calculate the output layer unit outputs:

$$\hat{y}_{ij} = S\left(\sum_k^Q wo_{jk} \hat{z}_{ik} + bo_j\right) \quad (j=1,2,\dots,P) \quad (7)$$

where  $wo$  is the weight and  $bo$  is the bias.

(3) Compute the differential coefficient of output unit errors:

$$eo_j = (y_{ij} - \hat{y}_{ij})(1 - \hat{y}_{ij})\hat{y}_{ij} \quad (j=1,2,\dots,P) \quad (8)$$

(4) Compute the differential coefficient of layer unit errors:

$$eh_j = \hat{z}_{ij}(1 - \hat{z}_{ij}) \sum_{k=1}^Q eo_k wo_{kj} \quad (j=1,2,\dots,H) \quad (9)$$

(5) Adjust the weights of hidden layer units:

$$wh'_{jk} = wh_{jk} + \alpha * eh_j * x_{ik} \quad (j=1,2,\dots,H, k=1,2,Q) \quad (10)$$

where  $\alpha$  is learning rate.

(6) Adjust the biases of hidden layer units:

$$bh'_j = bh_j + \alpha * eh_j \quad (j=1,2,\dots,H) \quad (11)$$

(7) Adjust the weights of output layer units:

$$wo'_{jk} = wo_{jk} + \alpha * eo_j * \hat{y}_{ik} \quad (j=1,2,\dots,P, k=1,2,H) \quad (12)$$

(8) Adjust the biases of output layer units:

$$bo'_j = bo_j + \alpha * eo_j \quad (j=1, 2,\dots, P) \quad (13)$$

In each step of training process, MSE is  $E_{\Delta}$  calculated as:

$$E_{\Delta} = \sqrt{\left(\frac{1}{R * P}\right) \sum_{i=1}^R \sum_{j=1}^P \left(\frac{y_{ij} - \hat{y}_{ij}}{y_{ij}}\right)^2} \quad (14)$$

$E_{\Delta}$  is used to identify the network convergence. As it getting minimal value or being less than specified value, the training process will be stopped.

### 3 Improved Training Algorithm of BPNN for Time Series Forecasting

In the traditional network training algorithm, the learning rate  $\alpha$  is a constant. All the training samples whether near to or far from present have the same contribution to the BPNN. As the time series forecasting particularity, it is known that the near samples have more contribution than the far samples, that means near samples are more important. So we put forward an improved training algorithm of dynamic learning rate. The learning rate is going up as the time-point going up when the training samples change.  $\alpha_i = f(i)$  ( $i=1, 2, \dots, R$ ),  $f(\cdot)$  can be linear or logarithmic increasing function. We give the scope of the learning rate by specified the maximal value  $\alpha_{\max}$  and minimal value  $\alpha_{\min}$ .

Then  $\alpha_i \in [\alpha_{\min}, \alpha_{\max}]$ ,  $0 < \alpha_{\min} \leq \alpha_{\max} < 1$ .

Linear increasing function:

$$\alpha_i = \frac{(\alpha_{\max} - \alpha_{\min})}{(R - 1)} * (i - 1) + \alpha_{\min} \tag{15}$$

Logarithmic increasing function:

$$\alpha_i = \frac{(\alpha_{\max} - \alpha_{\min})}{\ln(R)} * \ln(i) + \alpha_{\min} \tag{16}$$

In the dynamic learning rate training process, the errors should be transformed for identifying the network convergence. There are two ways to transform the errors corresponding to the two ways of dynamic learning rate. The  $j$  th output transformed error is corresponding to the  $i$  th training sample is:

$$e'_{ij} = \lambda_i e_{ij} = \lambda_i (y_{ij} - \hat{y}_{ij}) \tag{17}$$

where  $\lambda_i \in [\lambda_{\min}, 1]$ ,  $\lambda_i$  is corresponding to  $\alpha_i$ .

Linear transform:

$$\lambda_i = \frac{(1 - \lambda_{\min})}{(R - 1)} * (i - 1) + \lambda_{\min} \tag{18}$$

Logarithm Transform:

$$\lambda_i = \frac{(1 - \lambda_{\min})}{\ln(R)} * \ln(i) + \lambda_{\min} \tag{19}$$

The actual error is also transformed by using  $\lambda$  :

$$E'_\Delta = \sqrt{\left(\frac{1}{R * P}\right) \sum_{i=1}^R \sum_{j=1}^P \left(\frac{\lambda_i (y_{ij} - \hat{y}_{ij})}{y_{ij}}\right)^2} \tag{20}$$

$E'_\Delta$  reaches minimal value or is less than a specified value would be the standard to identify whether the network is convergence or not. This network training algorithm makes the network fit the far samples cursorily and fit the near samples accurately so as to increase the effectiveness of forecasts and reduce the training times.

### 4 Forecasting Cases Using Improved Training Algorithm

We developed an application programmed with C++ for time series forecasting based on BPNN, in the network training process we can choose the learning rate as constant, linear going up and logarithm going up. By using the application we build models and make forecasts for some of time series. To demonstrate the improved algorithm effectiveness and efficiency, we present two cases (including single-variable and a multi-variable time series). In each case, we can compare training times, sum of square errors of training  $E_t^2$  and sum square errors of forecasting  $E_f^2$  gotten by using the three training algorithms.

$$E_t^2 = \frac{1}{R} \sum_{i=1}^R \sum_{j=1}^P (y_{ij} - \hat{y}_{ij})^2 \tag{21}$$

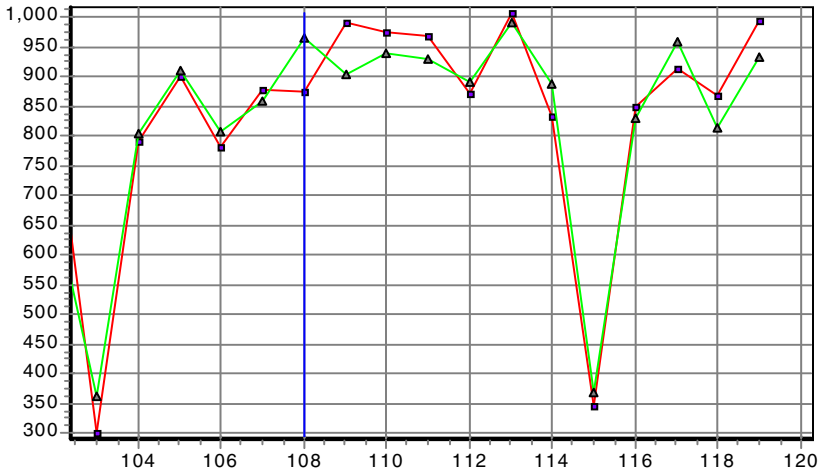
$$E_f^2 = \frac{1}{R} \sum_{i=1}^F \sum_{j=1}^P (y_{ij} - \hat{y}_{ij})^2 \tag{22}$$

Case 1 is a single-variable time series. The data are from page 433 of Makridaks [9]. There are 120 time-point paper sales values of a company recorded monthly for 10 years. We use the early 108 values for modeling and the late 12 values for forecasting check (F=12). We choose the lag M=12, that means using 12 values to compute the next value in the network. The number of input units is Q=12. The number of output unit is P=1. Then we get R=N-M=96 training samples. The network convergence mean square error is  $E_\Delta = E'_\Delta \leq 0.01$ . We use three training algorithms to train networks and make forecasts. The results are showed as Table 1.

We can find out that the MSE of forecasting is lowest. The forecasts curve is showed as Fig.2. Curve with ■ is actual time series, and Curve with ▲ is forecasts.

**Table 1.** Single-variable time series training and forecasting results

Learning rate		Training Cycle Times*	MSE of Training	MSE of forecasting
Constant	0.5	1831	3598.5144	2462.0889
	0.7	1663	3579.0325	2646.7976
	0.9	1599	3561.9824	2680.1248
Linear	(0.1-0.9)	1489	3266.4333	3001.4504
	(0.5-0.9)	1488	3257.0891	2311.4026
Logarithm	(0.1-0.9)	1425	3278.3406	2463.3191
	(0.5-0.9)	1428	3187.3398	2374.1438



**Fig. 2.** Forecasting curve by using linear going up learning rate BPNN

Generally, the network converges faster by using dynamic learning rate than by using constant learning rate.

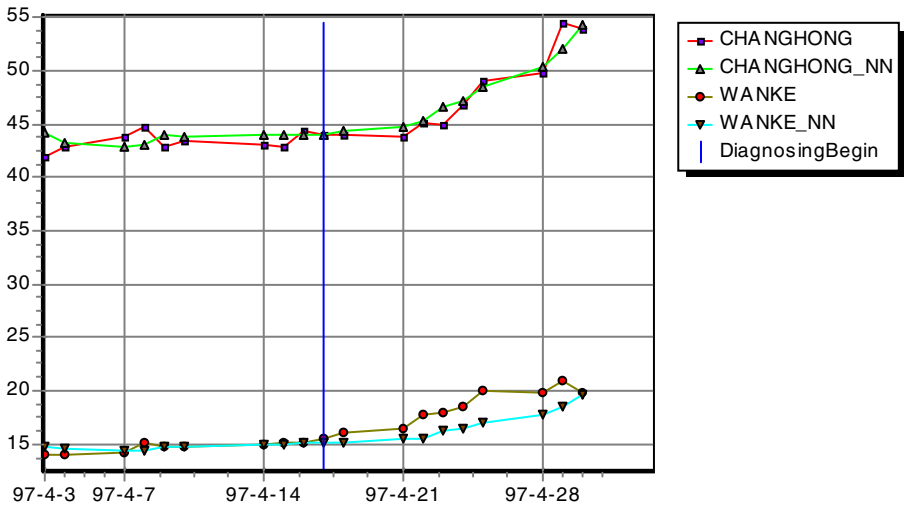
Case 2 is a multi-variable time series of Changhong and Wanke Stock prices from October 1996 to April 1997 in China, which has 126 time-point values. We use the early 116 values for modeling, the late 10 values for forecasting check. 10 values are used to compute the next 1 value in the network, the lag  $M=10$ . The number of output units  $P=2$ , the number of input units  $Q=P \times M= 20$ . Then there are  $R=N-M=106$  training samples. The mean square error for network convergence identification is  $E_{\Delta} = E'_{\Delta} \leq 0.01$ . we get the results as Table 2.

\* All training samples are used once to train network is called a cycle time.



**Table 2.** Multi-variable time series training and forecasting results

Learning rate		Training Cycle Times	MSE of Training	MSE of forecasting
Constant	0.5	4370	1.1916	22.2973
	0.7	2627	1.1548	22.8780
	0.9	1381	1.1500	24.4085
Linear	(0.1-0.9)	1142	3.1418	15.3209
	(0.5-0.9)	1089	2.7636	15.3335
Logarithm	(0.1-0.9)	1124	3.1333	15.4559
	(0.5-0.9)	1069	2.8207	15.0063



**Fig. 3.** Forecasting curve by using logarithm going up learning rate BPNN

From the results, we can find out that the network converge faster by using going up learning rate than by using constant learning rate. And the forecasts are more accurate. The forecasts are most accurate when the learning rate from 0.5 goes to 0.9 by logarithm mode. Its forecasting curve is showed as Fig.3. The forecasting values are those with \_NN in legend.

## 5 Conclusion

We propose an improved training algorithm of the learning rate dynamic going up with time points increasing for time series forecasting BPNN training. And we use the

transformed error to identify the network convergence. Those methods can make the network fit the early time point samples less accurate and fit the late time point samples more accurate. After comparing a lot of cases study results, we find that the proposed training algorithm for time series forecasting can decrease the training times and improve the forecasts veracity. Though the fitness for the training samples of dynamic learning rates is not better than that of constant learning rates in some time series cases, we can still get more accurate forecasts. This algorithm has particularly advantages in the modeling and forecasting for those time series with a large amount of training samples.

## References

1. Lapedes, A., Farber: Nonlinear signal processing using neural networks: Prediction and system modeling. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM (1987)
2. Werbos, P.J.: Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks* 1, 339–356 (1988)
3. Nam, K., Schaefer, T.: Forecasting international airline passenger traffic using neural networks. *Logistics and Transportation* 31, 239–251 (1995)
4. Shao, J.: Application of an artificial neural network to improve short-term road ice forecasts. *Expert Systems with Applications* 14, 471–482 (1998)
5. Tseng, F.M., Yu, H.C., Tzeng, G.H.: Combining neural network model with seasonal time series ARIMA model. *Technical forecasting & social change* 69, 71–87 (2002)
6. Wu, A., Hsieh, W.W., Tang, B.: Neural network forecasts of the tropical Pacific sea surface temperatures. *Neural networks* 19, 145–154 (2006)
7. Hill, T., O'Connor, M., Remus, W.: Neural network models for time series forecasts. *Management science* 42, 1082–1092 (1996)
8. Chiang, W.C., Urban, T.L., Baldrige, G.W.: A neural network approach to mutual fund net asset value forecasting. *Omega, International Journal of Management and Science* 24, 205–215 (1996)
9. Makridakis, S., et al.: *Forecasting: Methods and Applications*. John Wiley & Sons, Inc., Chichester (1983)

# Evolved Kernel Method for Time Series

Juan C. Cuevas-Tello\*

Engineering Faculty, Autonomous University of San Luis Potosí, México  
cuevas@uaslp.mx

**Abstract.** An evolutionary algorithm for parameter estimation of a kernel method for noisy and irregularly sampled time series is presented. We aim to estimate the time delay between time series coming from gravitational lensing in astronomy. The parameters to estimate include the delay, the width of kernels or smoothing, and a regularization parameter. We use mixed types to represent variables within the evolutionary algorithm. The algorithm is tested on several artificial data sets, and also on real astronomical observations. The performance of our method is compared with the most popular methods for time delay estimation. An statistical analysis of results is given, where the results of our approach are more accurate and significant than those of other methods.

**Keywords:** Evolutionary Algorithms, Genetic Algorithms, Kernel Methods, Machine Learning, Time Series.

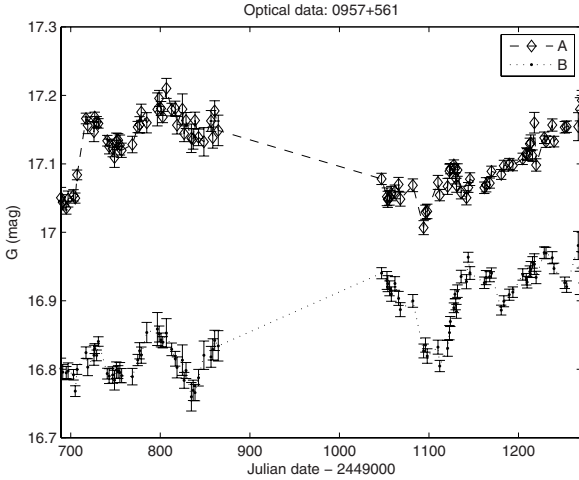
## 1 Introduction

On the one hand, kernel methods, which are the core ingredient of so popular support vector machines (SVM) [1], have been widely used in many application areas including pattern analysis, pattern recognition, classification problems and bio-informatics [2]. Radial Basis Function (RBF) networks are similar to kernels, but developed in a different field; i.e., artificial neural networks. In fact, a Gaussian kernel is sometimes referred to as RBF Kernel.

On the other hand, the algorithm to be presented here is an evolutionary algorithm (EA), which performs artificial evolution. This kind of algorithms belong to a new growing area, natural computation. Several evolutionary computational models have been proposed: *i*) genetic algorithms, *ii*) evolutionary programming, *iii*) evolution strategies and *iv*) genetic programming mainly. Other recent approaches have been introduced, but they are variants or improvements of the above evolutionary algorithms. These models share the common ingredient in evolution, that is the evolutionary operators such as selection, recombination and mutation. Some use either recombination or mutation, others put more emphasis on specific operators, but all of them are inspired by natural evolution. The EA proposed here comes from genetic algorithms (GAs) with real and integer representation; the typical representation in GAs is binary. Moreover, we compare it with an evolutionary strategy, (1+1)ES [3].

---

\* This paper was possible thanks to Peter Tiño, Somak Raychaudhury, Markus Harva and Xin Yao.



**Fig. 1.** Observations of the brightness of the doubly-imaged quasar Q0957+561, in the *g*-band, as a function of time (Top: Image A; Bottom: Image B). The time is measured in days (Julian days-2,449,000 days).

The EA is a data-driven approach. An example of the data (time series) is shown in Fig. 1.

There are several methods to deal with this kind of time series, including correlation-based methods [4,5], Gaussian processes [6], curve fitting [5] and Bayesian methods [7]. However, accurate estimations are still required [8]. The kernel method is chosen because in [9] is shown a superior performance of this method over correlation-based and Gaussian processes methods.

The contribution of this paper is the parameter optimization of a kernel method [9] for time series, which are irregularly sampled and noisy (see Fig. 1). Therefore, we present an EA for time delay estimation between such time series. Moreover, the regularization of the kernel method within the EA is also a novel approach.

The organization of this paper is as follows: first, the kernel method for time series is introduced along with the new regularization procedure, then, the EA is presented. In §4.2, we present the data for the experiments. At the end, we present the results and the statistical analysis.

## 2 Kernel Method

The kernel method for irregularly sampled time series is proposed by [9,10]. The aim of this section is to come up with the parameters to evolve.

Referring to Fig. 1, images A and B are modelled as a pair of time series

$$\begin{aligned} x_A(t_i) &= h_A(t_i) + \varepsilon_A(t_i) \\ x_B(t_i) &= h_B(t_i) \ominus M + \varepsilon_B(t_i), \end{aligned} \tag{1}$$

where  $\ominus = \{\times, -\}$  denotes either multiplication or subtraction, so  $M$  is either a ratio (used in radio observations, where brightness is quoted in flux units) or an offset between the two images (as in optical observations, where brightness is represented in logarithmic units). The latter option is used here. Values of the independent variable  $t_i, i = 1, 2, \dots, n$  represent discrete observation times. The observation errors  $\varepsilon_A(t_i)$  and  $\varepsilon_B(t_i)$  are modelled as zero-mean Normal distributions  $N(0, \sigma_A(t_i))$  and  $N(0, \sigma_B(t_i))$ , respectively, where  $\sigma_A(t_i)$  and  $\sigma_B(t_i)$  are standard deviations.

Now,

$$h_A(t_i) = \sum_{j=1}^N \alpha_j K(c_j, t_i) \tag{2}$$

is the ‘‘underlying’’ light curve that underpins image A, whereas

$$h_B(t_i) = \sum_{j=1}^N \alpha_j K(c_j + \Delta, t_i) \tag{3}$$

is a time-delayed (by  $\Delta$ ) version of  $h_A(t_i)$  underpinning image B.

The functions  $h_A$  and  $h_B$  are formulated within the generalized linear regression framework [11,2]. Each function is a linear superposition of  $N$  kernels  $K(\cdot, \cdot)$  centered at either  $c_j, j = 1, 2, \dots, N$  (function  $f_A$ ), or  $c_j + \Delta, j = 1, 2, \dots, N$  (function  $f_B$ ). Gaussian kernels of width  $\omega_c$  are used: for  $c, t \in \mathfrak{R}, K(c, t) = \exp \frac{-|t-c|^2}{\omega_c^2}$ .

The kernel width  $\omega_c > 0$  determines the ‘degree of smoothness’ of the models  $h_A$  and  $h_B$ . The kernels are located at the position of each observation, implying  $N = n$ . The variable width  $\omega_j \equiv \omega_c$  is determined through the  $k$  nearest neighbors of  $c_j$  (equal to  $t_j$ ) as  $\omega_j = \sum_{d=1}^k (t_{j+d} - t_{j-d})$ .

The weights  $\alpha$  (2)-(3) are obtained (or learnt) as follows (see [9]):

$$\alpha = \mathbf{K}^+ \mathbf{x}. \tag{4}$$

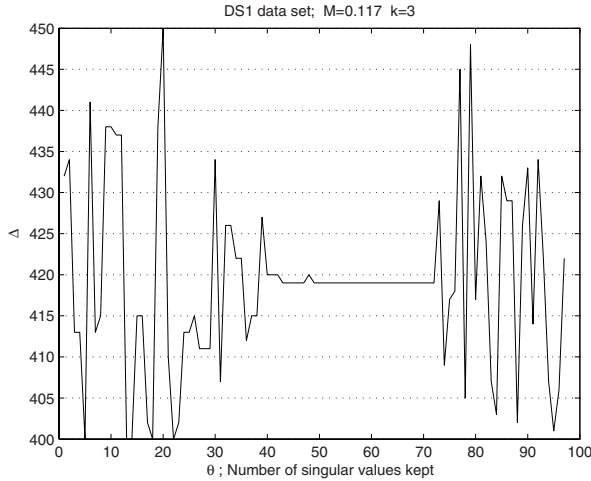
The aim is to estimate the time delay  $\Delta$  between the temporal light curves corresponding to images A and B. Typically,  $\Delta$  is estimated by a set of trial time delays in the range  $[\Delta_{min}, \Delta_{max}]$  with a specific measurement of goodness of fit [9].

Finally, the parameters are the time delay  $\Delta$  [as given in Eqs. (2) & (3)], the variable width  $k$ , and the regularization parameter [1]  $\lambda$ .

### 2.1 Regularization

In practice, the matrix  $\mathbf{K}$  (4) may be singular because  $\mathbf{K}$  is an overdetermined system, and noisy time series are involved, so singular value decomposition (SVD) is employed to regularize the inversion in (4) [11,9]. To avoid singularity, the most straightforward method is to find a threshold  $\lambda$  for singular values [11,9]. This means that the singular values less than  $\lambda$  are set to zero [11].

<sup>1</sup> This parameter is the singular value threshold, when computing  $\mathbf{K}^+$ .



**Fig. 2.** Patterns. In each relation  $(\Delta, \theta)$ , the best time delay has been plotted, i.e., the best  $\Delta$  ( $y$ -axis) versus the number of singular values  $\theta$  set to zero ( $x$ -axis). The best time delay is found through log-likelihood [9] by evaluating time delay trials in a given range. The data corresponds to the  $g$ -band optical observations of quasar Q0957+561;  $\Delta = [400, 450]$  with unitary increments;  $M$  was set to 0.117 [5] and  $k = 3$ . The pattern is at  $\theta = [49, 72]$ , where  $\Delta = 419$ .

In other words,  $\lambda$  tells us how many singular values to set to zero. Hence, for a given  $\Delta$ , the number of singular values to keep may vary. We illustrate this through Fig. 2, representing optical data, where  $\theta$  is the number of singular values to set to zero. One can see a well defined pattern in the range  $\theta = [49, 72]$  ( $\Delta = 419$ ) in Fig. 2. Thus, if one can find a proper  $\lambda$  that falls in this range, then one can claim that the estimation of  $\Delta$  is “robust”. But the range of this pattern may change for other  $M$  and  $k$  parameters, in which case there is no guarantee that the estimated  $\lambda$  falls in this range. Furthermore, no matter which method is used for assessing the goodness of fit, if we test  $\Delta$  in a specific range with a fixed  $\lambda$ , then we may come up with different values of  $\theta$  – some inside the pattern, some outside, none inside, etc.

Instead of  $\lambda$ , here  $\theta$  is employed as a regularization parameter in [3]. In fact, we aim to create an automatic algorithm that performs a global search for all parameters, and then finds the proper  $\theta$  that falls in the pattern; with our EA, this is done (see [3]).

### 3 Evolutionary Algorithm

Following the kernel method in [2], there are three parameters: i) the time delay  $\Delta$ , ii) the variable width  $k$  and iii) the amount of singular values to keep  $\theta$ . Besides, we have the fitting measurement; e.g., log-likelihood or any loss function. We follow an EA to avoid local minima [12].

We define as our population

$$P_1 = \{(\Delta_1, \theta_1, k_1), (\Delta_2, \theta_2, k_2), \dots, (\Delta_x, \theta_x, k_x), \dots, (\Delta_{n_p}, \theta_{n_p}, k_{n_p})\}, \quad (5)$$

where each element in  $P_1$  is a hypothesis commonly referred as individual or chromosome, which is a set of parameters  $\{\Delta_x, \theta_x, k_x\}$  initialized randomly. Then we have  $n_p$  hypotheses. Each hypothesis  $x$  is evaluated by  $f_x$  that is a measure of fitness pointing the best hypothesis out. Then, we apply artificial genetic operators such as selection, crossover, mutation and re-insertion (elitist strategy) to generate  $P_2, \dots, P_{n_g}$  populations. At the  $n_g$  generation, we choose from  $P_{n_g}$  the best set of parameters (or individual) according to its fitness; i.e., with minimum  $f_x$ . This process leads to artificial evolution, which is a stochastic global search and optimization method based on the principles of biological evolution [12].

For mixed types, we represent every population  $P_1$  to  $P_{n_g}$  as two linked populations of the same size  $n_p$ ,  $P_1 = [P_1^1 P_1^2]$ . Hence,  $P_1^1$  uses reals to represent  $\Delta_x$ , and  $P_1^2$  employs integers to represent  $\theta_x$  and  $k_x$ .

We employ a population size of  $n_p = 300$  individuals and  $n_g = 50$  generations unless other values are given. We use the Genetic Algorithm Toolbox<sup>2</sup> [13] for MATLAB®.

### 3.1 Fitness Function

We use as a measure of fitness (or objective function) the cross-validation procedure (CV) [1]. That is, the mean squared error (MSE) is given by CV as described below in Algorithm 1, where  $T = A - V$  is the training set;  $A$  is the set of all observations, and  $V$  is the validation set.

### 3.2 Evolution Operators

We use the basic roulette wheel **selection** method for both  $P_1^1$  and  $P_1^2$ , which is stochastic sampling with replacement. Consequently, this is a mechanism to probabilistically select individuals from a population based on their fitness function. The higher the fitness value, the larger the interval in the wheel [13]. From the initial population, we select half of the population so we work with this population during recombination, mutation and evaluation. Finally, we reinsert the best individuals to the initial population to obtain a population of size  $n_p$  for the next generation. In other words, we perform re-insertion of offsprings [13]. Other selection methods were tested such as tournament selection and stochastic universal sampling, but roulette wheel selection gave us the better results on artificial and real data.

Four methods for **recombination** (or crossover) have been tested: discrete, intermediate, linear and double-point recombination. The last one is only for integer representation. They all lead to similar results on the real data in §4.1 and on some artificial data sets so we adopt linear recombination for reals and

<sup>2</sup> Which is available online with a good documentation.

**Algorithm 1.** Fitness Function ( $A, \Delta_x, k_x, \theta_x$ )

---

```

/* A is the set of all observations; its cardinality is n          */
1 Fix Blocks  $\leftarrow 5$ 
2 Fix PointsPerBlock  $\leftarrow n/Blocks$ 
3 for  $l \leftarrow 1$  to PointsPerBlock do
4   Remove the  $l^{th}$  observation of each block and include it in  $V$ 
5   Compute  $\mathbf{h}_A$  and  $\mathbf{h}_B$  for the training set  $T = A - V$ 
6   Obtain  $MSE_{CV}$  on the validation set  $V$ 
7    $R(l) \leftarrow MSE_{CV}$ 
8  $f_x \leftarrow mean(R)$ 
9 return  $f_x$ 

```

---

double-point for integers. Linear recombination can generate an offspring on a slightly longer line than that defined by its parents. Whereas  $o = p_1 + \alpha \times (p_2 - p_1)$ , such as  $\alpha = U[-0.25, 1.25]$  is uniformly distributed, and  $o$  is the offspring with parents  $p_1$  and  $p_2$  [13]. Double-point recombination involves selecting uniformly at random two integer positions to exchange the variables in those positions. Typically this method is used for binary representation, but integers also can be used [13].

For reals, we tested two methods for **mutation**: Gaussian mutation and *mutbga* (as in Breeder Genetic Algorithm [13]). Both lead to similar performance. Therefore, we adopt *mutbga* as our mutation operator; whereas a mutated variable can be obtained by  $m_M = v + s_1 \times r \times s_2 \times \delta^M$ , where  $m_M$  is the mutated variable,  $v$  is the variable to mutate,  $s_1 = \pm 1$  with a probability given by a mutation rate in the range  $[0,1]$ ,  $r = 0.5 \times d$  ( $d$  is the domain of variable) and  $\delta^M = \sum_{i=0}^{m-1} \alpha_i^M 2^{-i}$ ;  $\alpha_i^M = 1$  with probability  $1/m$ , else  $0 - m = 20$ . For integers, we use 0.5 as mutation rate.

## 4 Data

### 4.1 Real Data

This paper focuses on optical data only. Optical astronomers measure the brightness of a source (flux) using imaging devices (e.g., Charge-Coupled Device – CCD), with filters to restrict the range of wavelength/frequency of light observed. The flux  $f$  of light from a source is expressed in logarithmic units known as magnitudes (mag), defined as  $mag = -2.5 \log_{10} f + constant$ , where  $f$  can be represented in mJy (milliJanskys). The errors on mag are mainly measurement errors (see error bars in Fig. 1), assumed to be zero-mean Gaussian. The green (g-) band represent measurements obtained with filters in the wavelength range 400–550 nm. The data is given in tabular form with 97 rows (samples) and with five columns: Time<sup>3</sup>  $t$ , Image A, Error A, Image B and Error B; shown in Fig. 1. The source was monitored nightly, but many observations were missed due to

<sup>3</sup> Julian date -  $2.44 \times 10^6$ .



cloudy weather and telescope scheduling. The big gap in Fig. 1 is an intentional gap in the nightly monitoring, since a delay of about 400 days was known *a priori*. Therefore, the peak in the light curve of image A, between 700 and 800 days, corresponds to the peak in that of image B between 1,100 and 1,200 days. This is the delay that we aim to estimate. The time delay between the signals depends on the mass of the lens, and thus it is the most direct method to measure the distribution of matter in the Universe, which is often dark [14].

## 4.2 Artificial Data

**DS-5.** In this data set, the true time delay is 5 days [10], and the true offset in brightness between image A and image B is  $M = 0.1$  mag. The intention here is to simulate optical observations as in Ovoldsen *et al.* [8]. We employ five underlying functions (shapes). These data sets are irregularly sampled with three levels of noise and gaps of different size as in [9,10]. We generated 50 realizations per level of noise only. Consequently, this yields 38,505 data sets, with 50 samples each.

**Harva Data.** These data sets, generated by a Bayesian model [7], simulate three levels of noise with 225 data sets per level of noise, where each level of noise represents the variance:  $0.1^2$ ,  $0.2^2$  and  $0.4^2$ . The data are irregularly sampled and the true time delay in all cases is 35 days.

## 5 Results

This section presents the results of the evolved kernel method on real and artificial data. We compare the performance of our EA, on the same data sets, against two of the most popular methods from the astrophysics literature: (a) Dispersion spectra method [4], and (b) the structure-function-based method PRH, [6]. In addition, we compare with the performance of a non-evolutionary kernel approach, based on kernels with variable width (K-V) [9,10]. Two versions of Dispersion spectra are used;  $D_1^2$  is free of parameters [4] and  $D_{4,2}^2$  has a decorrelation length parameter  $\delta$  involving only nearby points in the weighted correlation [4]. For the case of the PRH method, we use the image A from the data to estimate the structure function [6]. In addition, we compare EA against a Bayesian method on data sets generated by this Bayesian approach [7].

### 5.1 Real Data

The observational optical data is outlined in §4.1. We use the following general bounds:  $\Delta = [400, 450]$ ,  $k = [1, 15]$  and  $\theta = [1, n]$ . We use the reported value  $M = 0.117$  [5]. The results of ten runs (realizations) are given in Table 1. The set  $\{\Delta, M, \theta, k\}$  is the best solution (individual) at  $g = 50$  according to  $f_x$  (i.e.,  $MSE_{CV}$ ). The column *Convergence* shows at what generation a stability has been reached, i.e., from what generation the  $MSE_{CV}$  is constant.

Of the continuous optimization approaches, we tested one, (1+1)ES [3], which is based on the Gray-code neighborhood distribution, and uses real representation. We chose this (1+1)ES because our fitness function is costly, so one expects to require less fitness evaluations than our EA. Rowe *et al.* [3] have shown superior performance of their (1+1)ES over Improved Fast Evolutionary Programming (IFEP), on some benchmark problems, and on a real-world problem (medical tissue optics). IFEP is also a continuous optimization approach. For (1+1)ES, the precision is set to 200, and variable bounds set as above, allowing until 15,000 iterations. The convergence is reached after 14,410 iterations by using the same fitness function (Algorithm 1 in [3]), so we also floor at fitness function for integer variables. This ES yields  $\Delta = 419.6$ ,  $M = 0.1732$ ,  $\theta = 58$ ,  $k = 3$ , and  $MSE_{CV} = 1.9249617 \times 10^{-3}$ .

**Table 1.** Evolutionary algorithm with mixed types: results on real data

Run	$\Delta$	$\theta$	$k$	$f_x$	Convergence at
1	419.68	58	3	0.0019249744	42
2	419.67	58	3	0.0019249722	34
3	419.69	58	3	0.0019249722	47
4	419.67	58	3	0.0019249719	49
5	419.66	58	3	0.0019249691	40
6	419.66	58	3	0.0019249670	45
7	419.66	58	3	0.0019249753	44
8	419.67	58	3	0.0019249724	47
9	419.47	71	3	0.0018908716	32
10	419.67	58	3	0.0019249711	49

$\Delta$  is given in days

We point out that in Table 1 the EA suggests  $\theta = 58$ , which falls within the pattern in Fig. 2.

The results of the (1+1)-ES are also consistent, even though it requires a larger number of iterations. On the one hand, for our EA, if  $g = 50$  (maximum number of generations), then we perform 7,800 evaluations of the fitness function, because of our elitist strategy. On the other hand, (1+1)-ES converges for around 14,000 iterations for different initializations. Every iteration corresponds to a fitness evaluation. Therefore, (1+1)-ES demands more computational time (about twice as much), so we do not use this algorithm to analyze artificial data. Since we use the same fitness function, one expects to get similar performance to EA. Moreover, a theoretical analysis in multi-objective optimization suggests a better performance from population-based algorithms [15].

### 5.2 Artificial Data

**DS-5.** In all cases, the time delay under analysis is given by trials of values between  $\Delta_{min} = 0$  and  $\Delta_{max} = 10$  (also bounds in our EA), with increments of 0.1, where the ratio  $M$  is set to its true value 0.1. The parameter  $\delta$  is set to

**Table 2.** DS-5 results: t-test

Method	0%	0.036%	0.106%	0.466%
$D_1^2$	10	13	21	20
$D_{4,2}^2$	6	1	0	0
PRH	0	2	14	16
K-V	11	5	6	13
EA	<b>22</b>	<b>23</b>	<b>24</b>	<b>22</b>

see §5.2 for details

5, for  $D_{4,2}^2$ . When using the PRH method, we use bins in the range of  $[0, 10]$  for estimating the structure function from the light curve of Image A. In our EA, besides the above  $\Delta$  bounds, we use the following bounds:  $\theta = [1, n]$ , and  $k = [1, 15]$ . For K-V, we cross-validate  $k$  and  $\lambda$ ; the ranges are  $k = [1, 15]$  and  $\lambda = [10^{-1}, 10^{-2}, \dots, 10^{-6}]$  (see §2.1).

We performed a t-test on these results, where the hypothesis to test is  $H_0: \mu_0 = 5$ . Since  $\mathcal{T}$  follows a Student’s t-distribution, which is centered at zero, those values close to zero are statistically significant. We use the threshold  $\alpha = 0.05$  for a significance level of 95%. In Table 2, we show the number of cases that satisfy the above threshold values. The results are grouped by noise level, and the best ones are highlighted in bold. We also tested the significance of time delay estimates with non-parametric hypothesis testing, such as sign test and Wilcoxon’s signed-rank test, with similar results.

**Harva Data.** Similarly, we compare the performance of EA with another kind of data; see §4.2. These data come from a Bayesian approach [7]. We performed an analysis as above, where the hypothesis to test is  $H_0 : \mu_0 = 35$  (the true delay); i.e.,  $\mathcal{T}$  and  $\mathcal{P}$  values; MSE is the mean squared error, the true against the estimated value; AE is the average of the absolute error;  $\hat{\mu}$  is the mean and  $\hat{\sigma}$  the standard deviation of estimates.

**Table 3.** Harva Data results: Statistical Analysis

Method	Statistic	Data Set		
		0.1	0.2	0.4
Bayesian method	$\mathcal{P}$	0.59	0.63	<b>0.06</b>
	$\mathcal{T}$	0.52	0.48	<b>1.87</b>
	MSE	32.18	<b>9.43</b>	<b>41.89</b>
	AE	1.84	<b>1.94</b>	<b>3.7</b>
	$\hat{\mu}$	35.2	35.1	<b>35.8</b>
	$\hat{\sigma}$	5.7	<b>3.1</b>	<b>6.4</b>
EA	$\mathcal{P}$	<b>0.92</b>	<b>0.76</b>	0.007
	$\mathcal{T}$	<b>0.09</b>	<b>0.30</b>	2.70
	MSE	<b>10.06</b>	23.25	66.99
	AE	<b>1.76</b>	3.28	5.72
	$\hat{\mu}$	<b>35.0</b>	35.1	36.4
	$\hat{\sigma}$	<b>3.1</b>	4.8	8.0

Since the data were generated by the Bayesian method, we aim to compare such a method with EA only.

In Table 3, the EA results are more significant for data with noise levels of 0.1 and 0.2, where  $\mathcal{P}$  is 0.92 and 0.76 respectively. But for a noise level of 0.4, it does not perform as well. In terms of bias ( $\hat{\mu} - \mu_0$ ), EA performs better for the data with a noise level of 0.1, and ties in the case of that with 0.2 data and on the noise of 0.4 data, the performance is not good enough. Talking about variance  $\hat{\sigma}$ , EA is better on data set of noise 0.1 only. It needs to be emphasized though, that these data was generated by the Bayesian estimation method, so the comparison is positively biased towards the Bayesian method.

## 6 Conclusions

Regarding the real data, in the astrophysics literature, the best (smallest quoted error) previous measures for this time delay can be found to be  $417 \pm 3$  [5] and  $419.5 \pm 0.8$  [10]. Therefore, the results in Table 1 are consistent. However, we think that the estimate of  $417 \pm 3$  days, from this data set, is underestimated because, for the quasar Q0957+561, the latest reports also give estimates around 420 days by using other data sets [8]. One is reminded that the gravitational lensing theory predicts that the time delay must be the same regardless of the wavelength of observation [14]. Therefore, we conclude that the definitive value of the time delay for this optical data set is 419.6 days.

In Table 2, where results from DS-5 are grouped by noise level, the results from EA are more statistically significant than others. This is an important result from EA on DS-5, because its accuracy in this application is matched to the precision and low levels of noise with which the current state-of-the-art optical monitoring data are being acquired for multiple-image time delay measurement [8].

We conclude that for Harva data, EA outperforms the Bayesian estimation method only for low level of noise (0.1 data). In other cases, depending on the statistic, EA can show better, equal or worse results.

As future work, there are several research lines to follow including the sparsity of the kernel method to deal with large data sets (above one thousand observations per data set), the application of this method on real data from other quasars, the study of microlensing and supernova explosions since this methodology models the source of light, and compare the EA with others non evolutionary optimization approaches for parameter estimation; e.g., simulated annealing.

## References

1. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2001)
2. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)

3. Rowe, J., Hidovic, D.: An evolution strategy using a continuous version of the gray-code neighbourhood distribution. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 725–736. Springer, Heidelberg (2004)
4. Pelt, J., Kayser, R., Refsdal, S., Schramm, T.: The light curve and the time delay of QSO 0957+561. *Astronomy and Astrophysics* 305(1), 97–106 (1996)
5. Kundic, T., Turner, E., Colley, W., Gott-III, J., Rhoads, J., Wang, Y., Bergeron, L., Gloria, K., Long, D., Malhorta, S., Wambsganss, J.: A robust determination of the time delay in 0957+561A,B and a measurement of the global value of Hubble's constant. *Astrophysical Journal* 482(1), 75–82 (1997)
6. Press, W., Rybicki, G., Hewitt, J.: The time delay of gravitational lens 0957+561, I. Methodology and analysis of optical photometric data. *Astrophysical Journal* 385(1), 404–415 (1992)
7. Harva, M., Raychaudhury, S.: Bayesian estimation of time delays between unevenly sampled signals. In: IEEE International Workshop on Machine Learning for Signal Processing, pp. 111–122. IEEE Computer Society Press, Los Alamitos (2006)
8. Ovaldsen, J., Teuber, J., Schild, R., Stabell, R.: New aperture photometry of QSO 0957+561; application to time delay and microlensing. *Astronomy and Astrophysics* 402(3), 891–904 (2003)
9. Cuevas-Tello, J., Tiño, P., Raychaudhury, S.: How accurate are the time delay estimates in gravitational lensing? *Astronomy and Astrophysics* 454, 695–706 (2006)
10. Cuevas-Tello, J., Tiño, P., Raychaudhury, S.: A kernel-based approach to estimating phase shifts between irregularly sampled time series: an application to gravitational lenses. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 614–621. Springer, Heidelberg (2006)
11. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: *Numerical Recipes in C++: The Art of Scientific Computing*, 2nd edn. Cambridge University Press, Cambridge (2002)
12. Goldberg, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1989)
13. Chipperfield, A.J., Fleming, P.J., Pohlheim, H., Fonseca, C.M.: *Genetic Algorithm Toolbox for use with MATLAB*. Automatic Control and Systems Engineering, University of Sheffield. 1.2 edn. (1996), <http://www.shef.ac.uk/acse/research/ecrg/getgat.html>
14. Kochanek, C.S., Schechter, P.L.: The Hubble Constant from Gravitational Lens Time Delays. In: Freedman, W.L. (ed.) *Measuring and Modeling the Universe*, p. 117 (2004)
15. Giel, O., Lehre, P.: On the effect of populations in evolutionary multi-objective optimization. In: Keijzer, M., et al. (eds.) *Genetic and Evolutionary Computation Conference (GECCO)*, vol. 1, pp. 651–658. ACM Press, New York (2006)

# Using Ant Colony Optimization and Self-organizing Map for Image Segmentation

Sara Saatchi and Chih-Cheng Hung

School of Computing and Software Engineering  
Southern Polytechnic State University  
1100 South Marietta Parkway  
Marietta, GA 30060 USA  
{ssaatchi, chung}@spsu.edu

**Abstract.** In this study, ant colony optimization (ACO) is integrated with the self-organizing map (SOM) for image segmentation. A comparative study with the combination of ACO and Simple Competitive Learning (SCL) is provided. ACO follows a learning mechanism through pheromone updates. In addition, pheromone and heuristic information are normalized and the effects on the results are investigated in this report. Preliminary experimental results indicate that the normalization of the parameters can improve the image segmentation results.

## 1 Introduction

Image segmentation plays an essential role in the interpretation of various kinds of images. Image segmentation techniques can be grouped into several categories such as edge-based segmentation, region-oriented segmentation, histogram thresholding, and clustering algorithms [1]. The aim of clustering algorithms is to aggregate data into groups such that data in each group share similar features while data clusters are being distinct from each other. A problematic issue in image segmentation is detecting objects which might not have data pixels with similar spectral features. Therefore an image segmentation procedure that is merely relied on spectral features of an image is not always desirable. To overcome this problem the spatial information besides other spectral information of the data pixels should also be considered.

There are a number of techniques, developed for optimization, inspired by the behavior of natural systems [2] and other techniques [3]. Swarm intelligence has been introduced in the literature as an optimization technique [4]. The ACO algorithm was first introduced and fully implemented in [4] on the traveling salesman problem (TSP) which can be stated as finding the shortest closed path in a given set of nodes that passes each node once. The ACO algorithm which we focus on is based on a sequence of local moves with a probabilistic decision based on a parameter, called pheromone as a guide to the objective solution. There are algorithms that while they follow the cited procedure of ACO algorithm, they do not necessarily follow all the aspects of it, which we informally refer to as ant-based algorithm or simply ant algorithm. There are several ant-based approaches to clustering which are based on the stochastic behavior of ants in piling up objects [5, 6, 7, 8, 9].

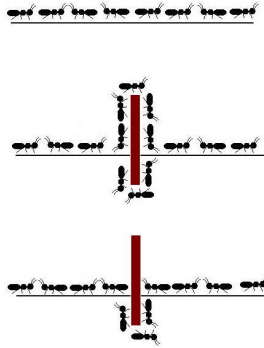
Self-Organizing Map (SOM) is a one layer neural network model. It was first introduced by Kohonen [10]. The SOM is a prominent unsupervised neural network model providing a topology preserving mapping from a high-dimensional input space to a two-dimensional feature map. Each neuron in the SOM is connected to the neighboring neuron which is different from the simple competitive learning neural network model. SOM has been used for dimension reduction and clustering in literature [11].

Competitive learning introduced in [12] is an interesting and powerful learning principle. It has been applied to many unsupervised learning problems. Simple competitive learning (SCL) is one of the several different competitive learning algorithms proposed in the literature. It shows the stability in data clustering applications over different run trials, but this stable result is not always the global optima. In fact, in some cases SCL converges to local optima over all run trials and the learning rate need to be adjusted in the course of experimentation so that the global optimization can be achieved. We have integrated the simple competitive learning algorithm with the ACO algorithm in [13]. In this paper we will study the integration of the self-organizing map algorithm with the ACO algorithm and compare the results with ACO-SCL algorithm. The reason is that this unsupervised neural network model has been widely used for data clustering [11]. Although SOM and SCL models are similar, it is well known that only a small number of neurons might be involved in the learning process for the SCL. Our purpose is to compare the impact of the ACO algorithm on these two models and their difference. Also, in this paper, pheromone and heuristic information in both ACO-SCL and SOM-SCL were normalized and the effect of this normalization on SCL and SOM algorithms was investigated.

## 2 Ant Colony Optimization (ACO)

The ACO heuristic has been inspired by the observation of real ant colony's foraging behavior and the fact that ants can often find the shortest path when searching for food. This is achieved by a deposited and accumulated chemical substance called pheromone by the passing ant which goes towards the food. In its searching the ant uses its own knowledge of where the smell of the food comes from (we call it as heuristic information) and the other ants' decision of the path toward the food (pheromone information). After it decides its own path, it confirms the path by depositing its own pheromone making the pheromone trail denser and more probable to be chosen by other ants. This is a learning mechanism ants possess besides their own recognition of the path. As a result of this consultation with the ants' behaviors already shown in searching the food and returning to the nest, the best path which is the shortest is marked from the nest towards the food.

In the literature [4], it was reported that the experiments show when the ants have two or more fixed paths with the same length available from nest to the food, the ants eventually concentrate on one of the paths and when the available paths are different in length they often concentrate on the shortest path. This is shown in Figure 1, when an obstacle is placed on the established path of ants, the ants first wander around the obstacle randomly. The ants going on a shorter path reach the food and return back to the nest more quickly. As the time passes on, the shorter path is reinforced by pheromone and eventually becomes the preferred path of the ants.



**Fig. 1.** Ants find the shortest path around an obstacle as a result of pheromone concentration

ACO uses this learning mechanism for the optimization. Furthermore, in the ACO algorithm, the pheromone level is updated based on the best solution obtained by a number of ants. The pheromone amount deposited by the succeeded ant is defined to be proportional to the quality of the solution it produces. For the real ants, the best solution is the shortest path. This path is marked with a strong pheromone trail. In the short path problem using the ACO algorithm, the pheromone amount deposited is inversely proportional to the length of the path. For a given problem the pheromone can be set to be proportional to any criteria of the desired solution. In the clustering method we introduced the criteria which include the similarity of data in each cluster, distinction of the clusters and compactness of them.

### 3 Self-organizing Map

SOM [10] has a topology similar to simple competitive learning (SCL) model except without the use of a neighborhood. The SOM consists of one layer of output nodes which are connected to each input node. Each output node includes a vector of weight with the same dimensionality as the input nodes. Each dimension of the weight vector is assumed to be a connection from the corresponding node to each dimension of the input node.

The algorithm starts by randomly initializing all the weights corresponding to output nodes. A sample set of inputs is used for training. Each training input is compared with the weight for each node in the layer and the node with the closest distance is selected as the best matching unit (BMU) or winner. The weight vectors of the BMU’s neighboring nodes are then updated such that the nodes closer to the BMU get changed more. In fact the weights of nodes at the boundary of the neighborhood window are barely changed. In general, the weight  $W(t+1)$  is updated by the learning formula given below:

$$W(t+1) = W(t) + \theta(t) L(t) (V(t) - W(t)) L(t) \tag{1}$$

where  $L$  is the learning rate which decays with time,  $V(t)$  is an input data vector and  $t$  refers to iteration time. The parameter  $L(t)$  can be defined as:



$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right) \quad (2)$$

where  $L_0$  denotes the learning rate at time  $t_0$ ,  $\lambda$  denotes a time constant,  $t$  refers to iteration and  $\theta$  defines the relation between distance of the nodes from the BMU and the influence on their learning. The parameter  $\theta$  is defined as:

$$\theta(t) = \exp\left(-\frac{dist^2}{2\sigma^2(t)}\right) \quad (3)$$

where  $dist$  is the distance of a node from the BMU and  $\sigma$  is the radius of the neighborhood. The radius of the neighborhood shrinks over time which causes  $\theta$  to decay over time. The parameter  $\sigma$  can be shrunk according to the following formula:

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda}\right) \quad (4)$$

where  $\sigma_0$  denotes the radius of the neighborhood at time  $t_0$ ,  $\lambda$  denotes a time constant and  $t$  refers to iteration.

## 4 The ACO and Simple Competitive Learning

Simple competitive learning is sometimes called a 0-neighbor Kohonen algorithm. It can be considered as a special case of SOM where the size of neighborhood window is reduced to one. The topology of the simple competitive learning algorithm can be represented as a one-layered output neural net. Each input node is connected to each output node. The number of input nodes is determined by the dimension of the training patterns. Unlike the output nodes in the Kohonen's feature map, there is no particular geometrical relationship between the output nodes in the simple competitive learning. In the following development, a 2-D one-layered output neural net will be used.

The algorithm ACO-SCL is described as follows. Let  $L$  denote the dimension of the input vectors, which for us is the number of spectral images (bands). We assume that a 2-D ( $N \times N$ ) output layer is defined for the algorithm, where  $N$  is chosen so that the expected number of the classes is less than or equal to  $N^2$ . Here weights of the nodes contain cluster center values.

Step 1: Initialize the number of clusters to  $K$  and the number of ants to  $m$ . Initialize pheromone level assigned to each pixel to 1 so that it does not have effect on the probability calculation in the first iteration.

Step 2: Initialize  $m$  sets of  $K$  different random cluster centers to be used by  $m$  ants.

Step 3: For each ant, assign each pixel  $X_n$  to one of the clusters ( $i$ ), randomly, with the probability distribution  $P_i(X_n)$  given in:

$$P_i(X_n) = \frac{[\tau_i(X_n)]^\alpha [\eta_i(X_n)]^\beta}{\sum_{j=0}^K [\tau_j(X_n)]^\alpha [\eta_j(X_n)]^\beta} \tag{5}$$

where  $P_i(X_n)$  is the probability of choosing pixel  $X_n$  in cluster  $i$ ,  $\tau_i(X_n)$  and  $\eta_i(X_n)$  are the pheromone and heuristic information assigned to pixel  $X_n$  in cluster  $i$  respectively,  $\alpha$  and  $\beta$  are constant parameters that determines the relative influence of the pheromone and heuristic information, and  $K$  is the number of clusters. Heuristic information  $\eta_i(X_n)$  is obtained from:

$$\eta_i(X_n) = \frac{\kappa}{CDist(X_n, CC_i) * PDist(X_n, PC_i)} \tag{6}$$

where  $X_n$  is the  $n$ th pixel,  $CC_i$  is the  $i$ th spectral cluster center, and  $PC_i$  is the  $i$ th spatial cluster center.  $CDist(X_n, CC_i)$  is the spectral Euclidean distance between  $X_n$  and  $CC_i$ , and  $PDist(X_n, PC_i)$  is the spatial Euclidean distance between  $X_n$  and  $PC_i$ . Constant  $\kappa$  is used to balance the value of  $\eta$  with  $\tau$ .

Step 4: For each input pixel the center of the cluster to which this input pixel belongs is considered as the BMU. Both spectral and spatial cluster centers are updated using:

$$C_i(t + 1) \leftarrow C_i(t) + \Delta(t)(x_i - C_i(t)), i = 1, \dots, L, \tag{7}$$

where  $\Delta(t)$  is a monotonically slowly decreasing function of  $t$  and its value is between 0 and 1.

Step 5: Save the best solution among the  $m$  solutions found. Our criteria for best solution include the similarity of data in each cluster, distinction of the clusters and compactness of them [14].

Step 6: Update the pheromone level on all pixels according to the best solution. The pheromone value is updated according to Eq. 8:

$$\tau_i(X_n) \leftarrow (1 - \rho) \tau_i(X_n) + \sum_i \Delta \tau_i(X_n) \tag{8}$$

where  $\rho$  is the evaporation factor ( $0 \leq \rho < 1$ ) which causes the earlier pheromones vanish over the iterations.

$\Delta \tau_i(X_n)$  in (8) is the amount of pheromone added to previous pheromone by the successful ant, which is obtained from:

$$\Delta \tau_i(X_n) = \begin{cases} \frac{Q * Min(k')}{AvgCDist(k', i) * AvgPDist(k', i)} & \text{if } X_n \text{ is a member of cluster } i. \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

In (9),  $Q$  is a positive constant which is related to the quantity of the added pheromone by ants,  $Min(k')$  is the maximum of the minimum distance between every two cluster centers obtained by ant  $k'$ ,  $AvgCDist(k', i)$  is the average of the spectral Euclidean distances within cluster  $i$  and  $AvgPDist(k', i)$  is the average of the spatial Euclidean distances, between all pixels in a cluster  $i$  and their cluster center obtained by ant

$k'$ .  $Min(k')$  causes the pheromone become bigger when clusters get more apart and hence raise the probability.

Step 7: Assign cluster center values of the best clustering solution to the clusters centers of all ants.

Step 8: If the termination criterion is satisfied go to next step. Otherwise, go to Step 3.

Step 9: Output the optimal solution.

## 5 The ACO and Self-organizing Map

The integrated algorithm ACO-SOM is described as follows. Let  $L$  denote the dimension of the input vectors, which for us is the number of spectral images (bands). We assume that a 2-D ( $N \times N$ ) output layer is defined for the algorithm, where  $N$  is chosen so that the expected number of the classes is less than or equal to  $N^2$ . Here weights of the nodes contain cluster center values.

Step 1: Initialize the number of clusters to  $K$  and the number of ants to  $m$ . Initialize pheromone level assigned to each pixel to 1 so that it does not have effect on the probability calculation in the first iteration.

Step 2: Initialize  $m$  sets of  $K$  different random cluster centers to be used by  $m$  ants.

Step 3: For each ant, assign each pixel  $X_n$  to one of the clusters ( $i$ ), randomly, with the probability distribution  $P_i(X_n)$  given in Eq. 5.

Step 4: For each input pixel the center of the cluster to which this input pixel belongs is considered as the BMU. Both spectral and spatial cluster centers are updated using Eq. 7. All other cluster center nodes within the neighboring window of the BMU are updated according to:

$$C_i(t+1) \leftarrow C_i(t) + \theta(t)\Delta(t)(x_i - C_i(t)), i = 1, \dots, L, \quad (10)$$

where  $\Delta(t)$  is a monotonically slowly decreasing function of  $t$  and its value is between 0 and 1. The parameter  $\theta$  is obtained from Eq. 3 where  $dist$  is the distance between neighboring cluster centers and the best matching cluster center.

Step 5: Save the best solution among the  $m$  solutions found. Our criteria for best solution include the similarity of data in each cluster, distinction of the clusters and compactness of them [14].

Step 6: Update the pheromone level for all pixels according to the best solution. The pheromone value is updated according to Eq. 8.

Step 7: Assign cluster center values of the best clustering solution to the cluster centers of all the ants.

Step 8: If the termination criterion is satisfied go to next step. Otherwise, go to Step 3.

Step 9: Output the optimal solution.

### 5.1 Normalized Pheromone and Heuristic Information

The pheromone and heuristic parameters were normalized and the effect of this normalization was investigated. Normalization of the pheromone assigned to a pixel in a cluster  $\tau_i(X_n)$  is performed such that the summation of the pheromone of all clusters equals to one. (i.e.  $\sum_{i=0}^c \tau_i(X_n) = 1$ , where  $c$  is the total number of clusters). This is the same for the heuristic information  $\eta_i(X_n)$ .

Normalization of the parameters prevents the values from getting too large so that only the relative influence of the parameters assigned to clusters is considered.

## 6 Simulation Results

For classification, the pixel-based minimum-distance classification algorithm was used in our experiments. Since SCL is very dependent on the learning rate, i.e.  $\Delta(t)$  in Eqs. 7 and 10, we performed some experiments on  $\Delta(t)$ . Considering that  $\Delta(t)$  is a monotonically slowly decreasing function of  $t$  and its value is between 0 and 1, we suggest the following formula:

$$\Delta(t) = \frac{0.2}{t^r + 1} \quad (11)$$

where  $t$  and  $r$  denote iteration and a rate which is a constant determined in the experiments. The experiments were performed over 20 run trials on two different images, for  $r$  from 10 to 50 incrementing by 10. Experiments showed that the better results were obtained for  $r = 10$ . Therefore the experiment was repeated similarly but for  $r$  from 1 to 10 incrementing by 1. This experiment showed that the better results were obtained for  $r$  between 1 and 5. In our experiments for the images shown in Figures 2 and 3,  $r$  was set to 2.

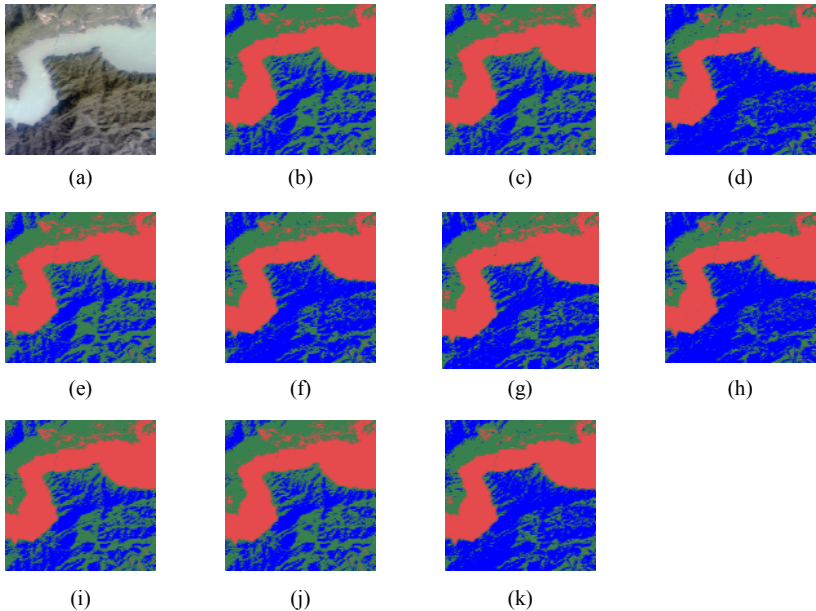
ACO-SCL algorithm showed that it is dependant on the set of parameters. Parameters used in ACO-SCL, other than  $r$  including  $\kappa$ ,  $Q$ ,  $\rho$ ,  $\alpha$ , and  $\beta$ . Parameters  $\alpha$ ,  $\beta$  and  $\kappa$  are used to keep the values of  $\tau$  and  $\eta$  in the same order. Parameter  $Q$  controls the added amount of pheromone and  $\rho$  eliminates the influence of the earlier added pheromone. Evaporation factor was set to be  $\rho = 0.8$ . We performed a course of experiments on the remaining parameters. Parameters  $\kappa$  and  $Q$  showed to have little influence on the results, while  $\alpha$  and  $\beta$  were more influential. The values tested were listed as follows:  $\kappa = 1000$  and  $10000$ ,  $Q = 10$  and  $100$ ,  $\alpha = 0.1$  to 50 incrementing by 10 and  $\beta = 0.1$  to 50 incrementing by 10. Each experiment was done for 20 run trials on different images. The results are not satisfactory with  $\beta = 0.1$  for images tested. The results are good with  $\alpha = 0.1$  for images tested but unstable. There were some set of

parameters that still did well for some of the images but not for the others. Knowing that  $\alpha$  should be small while  $\beta$  should not be small, we set up another experiment:  $\kappa = 1000$  and  $10000$ ,  $Q = 10$  and  $100$ ,  $\alpha = 0.1$  to  $2$  incrementing by  $0.1$  and  $\beta = 50$  to  $5$  decrementing by  $5$ . All the results were acceptable but not all were stable. So in this experiment stability of the results were examined. Experiment results show that  $\beta$  should not be very large otherwise it becomes unstable. When  $\beta$  is chosen to be  $5$  and  $\alpha$  is between  $0.1$  and  $2$ , the result showed to be more stable. So from these sets of experiment parameters were chosen as follows:  $r = 2$ ,  $\alpha = 2$ ,  $\beta = 5$ ,  $\kappa = 1000$ , and  $Q = 10$ . The number of ants was chosen to be  $m = 10$ .

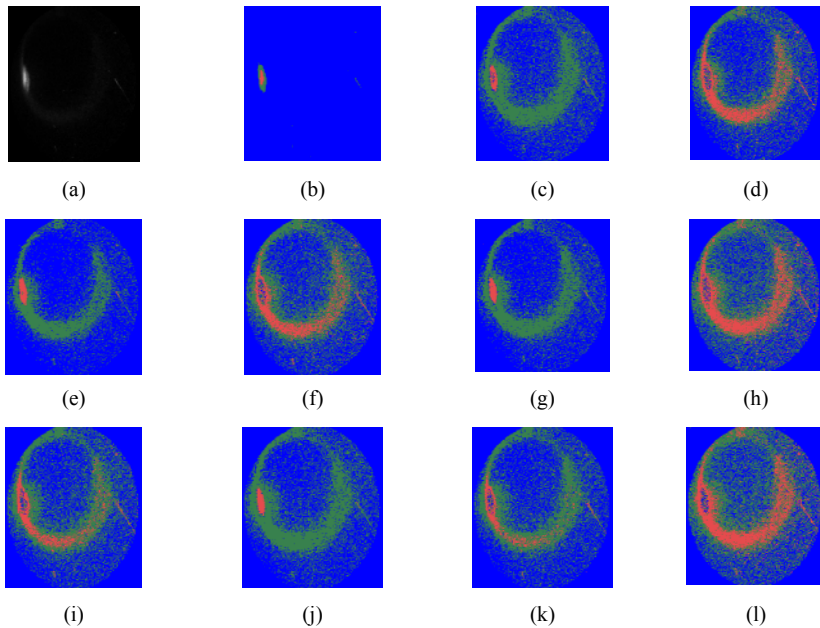
Some of the experimental results are shown in figures 2 and 3. Results on Aurora image clearly show that ACO can improve SCL in cases where the SCL algorithm is trapped into local optima. In order to further investigate on these algorithms, experiments were repeated 10 times on Aurora. Then the number of run trials that lead to

**Table 1.** Experimental results showing the number of runs that lead to global results out of 10 different run trials. Experiments are executed on the Aurora image.

	Parameters Are Not Normalized	Parameters Are Normalized
ACO-SCL	4	5
ACO-SOM	4	7



**Fig. 2.** Experimental results for river image compared with four algorithms; (a) original, (b) SCL, (c & d) ACO-SCL, (e & f) ACO-SCL with normalized parameters, (g, h, & i) ACO-SOM, and (j & k) ACO-SOM with normalized parameters. Results are obtained over 20 runs of each algorithm. Possible results on each run trial are shown.



**Fig. 3.** Experimental results for aurora image compared with four algorithms; (a) original, (b) SCL, (c & d) ACO-SCL, (e & f) ACO-SCL with normalized parameters and (g, h & i) ACO-SOM, (j, k, & l) ACO-SOM with normalized parameters. Results are obtained over 20 runs of each algorithm. Possible results on each run trial are shown.

the global optima were then counted. The results are shown in Table 1. As it can be seen the normalization of parameters has improved SOM in finding global optima.

## 7 Conclusion

In general the ACO can bring an advantage to the algorithms that are unstable resulting from dependency on random initialization. Although the simple competitive learning algorithm shows a high stability, the ACO still can make a contribution on the improvement of segmentation. The stable results of the SCL are not always the global optima and the ACO can be beneficial to SCL in finding the global optima. The ACO-SCL algorithm uses the same parameter set and learning rate as those used in SCL and recognizes the clusters where the SCL fails to do. This can be advantageous since for SCL to find the global optima the learning rate should be adjusted in the course of experimentation. Our observation on ACO-SOM algorithm showed similar effect. When the parameters were normalized, it was observed that the number of runs that lead to global results was improved. The ACO-SOM algorithm with normalized parameter showed to be more probable for finding the global solution in comparison to the ACO-SCL and ACO-SOM without normalization, and normalized ACO-SCL algorithms.

## References

1. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Addison-Wesley, Reading (1992)
2. Pham, D.T., Karaboga, D.: *Intelligent Optimization Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. Springer, Heidelberg (2000)
3. Hung, C.C., Scheunders, P.M., Pham, M.-C.S., Coleman, T.: Using Intelligent Optimization Techniques in the K-means Algorithm for Multispectral Image Classification. *Int. J. Fuzzy Syst.* 6(3), 105–115 (2004)
4. Dorigo, M., Maniezzo, V., Colomi, A.: Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cyb. Part B* 26, 29–41 (1996)
5. Yuqing, P., Xiangdan, H., Shang, L.: The K-means clustering Algorithm Based on Density and Ant Colony. In: *International Conference on Neural Networks and Signal Processing*, Nanjing, China, vol. 1, pp. 457–460 (2003)
6. Monmarché, N.: On data clustering with artificial ants. In: Freitas, A.A. (ed.) *AAAI-1999 and GECCO-1999 Workshop on Data Mining with Evolutionary Algorithms: Research Directions*, Orlando, Florida, pp. 23–26 (1999)
7. Monmarché, N., Slimane, M., Venturini, G.: AntClass: discovery of clusters in numeric data by an hybridization of ant colony with k-means algorithm, Internal Report no. 213 Laboratoire d'Informatique, E3i, Université de Tours, Tours, France (1999)
8. Bin, W., Yi, Z., Shaohui, L., Zhongzhi, S.: CSIM: A Document Clustering Algorithm Based on Swarm Intelligence. In: *Congress on Evolutionary Computation*, Honolulu, HI, vol. 1, pp. 477–482 (2002)
9. Kanade, P.M., Hall, L.O.: Fuzzy Ants as a Clustering Concept. In: *Proceedings of 22nd International Conference of the North American Fuzzy Information Processing Society*, Chicago, IL, pp. 227–232 (2003)
10. Kohonen, T.: The Self-Organizing Map. *Proceedings of the IEEE* 78(9), 1464–1480 (1990)
11. Vesanto, J., Alhoniemi, E.: Clustering of the Self-Organizing Map. *IEEE transaction on Neural Networks* 11(3) (2000)
12. Rumelhart, D.E., Zipser, D.: Feature discovery by competitive learning. In: McClelland, J.L., Rumelhart, D.E. (eds.) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pp. 151–193. MIT Press, Cambridge (1986)
13. Saatchi, S., Hung, C.C., Kuo, B.C.: A comparison of the improvement of k-means and simple competitive learning algorithms using ant colony optimization. In: *7th International Conference on Intelligent Technology*, Taipei, Taiwan (2006)
14. Saatchi, S., Hung, C.C.: Hybridization of the Ant Colony Optimization with the K-means Algorithm for Clustering. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) *SCIA 2005*. LNCS, vol. 3540, pp. 511–520. Springer, Heidelberg (2005)

# Correspondence Regions and Structured Images

Alberto Pastrana Palma and J. Andrew Bangham

University of East Anglia  
Norwich, NR47TJ, United Kingdom  
`{a.palma, ab}@uea.ac.uk`  
<http://www.uea.ac.uk>

**Abstract.** Finding correspondence regions between images is fundamental to recovering three dimensional information from multiple frames of the same scene and content based image retrieval. To be good, correspondence regions should be easily found, richly characterised and have a good trade-off between density and uniqueness. Maximally stable extremal regions (MSER's) are amongst the best known methods to tackle this problem. Here, we present an implementation of the sieve algorithm that not only generates MSER's but can also generate stable salient contours (SSC's) in different ways. The sieve decomposes the image according to local grayscale intensities and produces a tree in nearly  $O(N)$  where  $N$  is the number of pixels. The exact shape of the tree depends on the criteria used to control the merging of extremal regions with less extreme neighbours. We call the resulting data structure a 'structured image'. Here, a structured image in which MSER's are embedded is compared with those associated with two types of SSC's. The correspondence rate generated by each of these methods is compared using the standard evaluation method due to Mikalajczyk and the results show that SSC's identified using colour and texture moments are generally better than the others.

**Keywords:** Correspondance points, stable regions, sieve algorithm, connected-sets.

## 1 Introduction

Work in the fields of stereo matching and image retrieval has emphasised the importance of finding reliable correspondence points and regions. These should be easily identified, characterised by local features and rapidly compared with regions in other images. Ideally regions in the first image will uniquely match regions in the others. Usually, however, a cloud of candidate correspondences have to be searched for the correct matches. For image matching, Lowe [7], finds local extremal points in Gaussian scale space that are invariant to rotation and scale and robust in the face of some affine distortion, noise and changes in illumination. It is called the Scale Invariant Feature Transform (SIFT) and it has become a landmark for keypoint detection. In the other hand, for region detection, Matas developed Maximally Stable Extremal Regions (MSER's) [8]



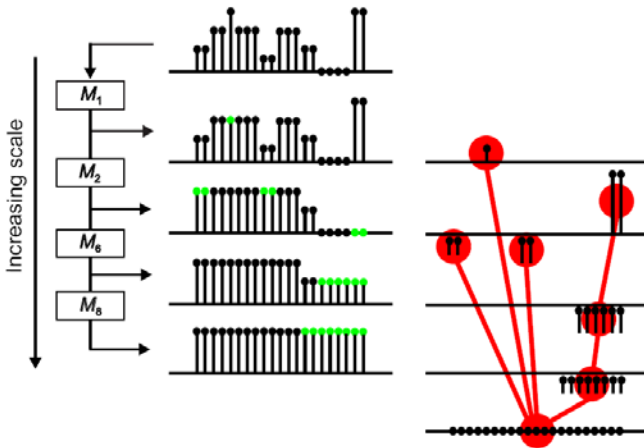
that have been reported to have good performance compared to other region detectors [9]. This too works in scale space as appointed by Lan et al. [5]. At this point a distinction between keypoints detectors like SIFT and region detectors like MSER's has to be established as they are not directly comparable to each other. In this paper we limit ourselves to region detection.

The sieve, developed by Bangham [2] in 1996, is a scale-space preserving algorithm that subsumes elements of MSER's. It has properties that can be controlled by the choice of operator that it uses at each scale and has been used for segmentation [1], texture analysis [12] and image retrieval [3]. Stable Salient Contours (SSCs) [5] make explicit use of the sieve algorithm to generate a set of regions for Stereo Matching that are a promising alternative to MSERs.

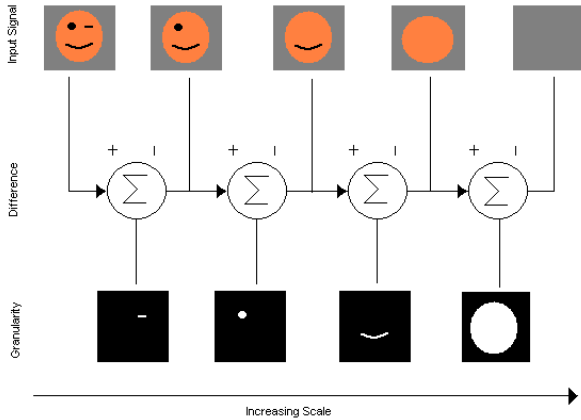
This paper presents a method for generating SSCs and shows how the associated tree could be used to accumulate a rich set of region descriptors.

## 2 The Sieve

The sieve algorithm applies connected-set morphological operators, specifically openings, closings or a combination of both to filter an input signal by removing local extrema throughout the scale-space. Flat extremal regions of a fixed area but variable shape are detected and merged with their neighbours that have the next most extreme values. This ensures that no new extrema can be created, i.e. it produces a scale space preserving decomposition. Figure 1 illustrates this process.



**Fig. 1.** Decomposition of a 1-D signal using the an  $M$  sieve. The left column shows the signal at different stages. The right column shows the differences between successive filtering stages (granules) and how these are linked together to generate a tree structure.



**Fig. 2.** Decomposition of a computer-generated image using the sieve algorithm. The top row shows images simplified with a closing ( $\mathcal{C}$ )-sieve. The bottom row shows the differences between successive filtering stages (granules),  $G_1^1, G_2^2, G_3^3, G_4^4$ .

Sieves are often described as operations over a graph  $G = (V, E)$  where  $V$  is the set of vertices that label the pixels and  $E$  are edges that indicate the adjacencies between pixels ([2] has definitions and references to graph morphology in which this notation is standard). There are four types of sieves corresponding to four grayscale graph morphology operators of open,  $\mathcal{O}$ , close,  $\mathcal{C}$ ,  $\mathcal{M}$ - and  $\mathcal{N}$ -filters.  $\mathcal{O}_s, \mathcal{C}_s, \mathcal{M}_s$  and  $\mathcal{N}_s$  are defined for each integer scale,  $s$ , at pixel,  $x$  as:

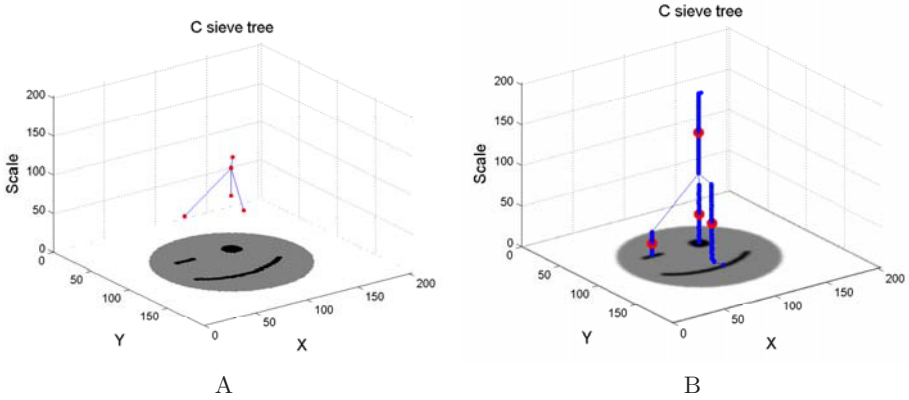
$$\mathcal{O}_s f(x) = \max_{\xi \in \mathcal{C}_s(G, x)} \min_{u \in \xi} f(u), \quad \mathcal{C}_s f(x) = \min_{\xi \in \mathcal{C}_s(G, x)} \max_{u \in \xi} f(u), \quad (1)$$

and  $\mathcal{M}_s = \mathcal{O}_s \mathcal{C}_s, \mathcal{N}_s = \mathcal{C}_s \mathcal{O}_s$ . Thus  $\mathcal{M}_s$  is an opening followed by a closing, both of size  $s$  and in any finite dimensional space. The sieves of a function,  $f \in \mathbf{Z}^V$  are defined in as sequences  $(f_s)_{s=1}^\infty$  with

$$f_1 = \mathcal{P}_1 f = f, \text{ and } f_{s+1} = \mathcal{P}_{s+1} f_s \quad (2)$$

for,  $s \geq 1$  where  $\mathcal{P}_s$  is one of  $\mathcal{O}_s, \mathcal{C}_s, \mathcal{M}_s$  and  $\mathcal{N}_s$ .

Sieves decompose an input image and remove its features by increasing the analysis scale from areas of size one (one pixel) to the size of the image area. Figure 2 shows a synthetic image that is progressively simplified by the removal of maxima and minima. The right “eye” is slightly larger than the left which is why it is removed at a larger scale. Here an  $\mathcal{C}$ -filter is used so the maxima and minima are removed together at each scale but, if the operator were an opening, then the image would be simplified by the progressive removal of maxima. Differences between successive scales are called *granule functions*, and these functions which exist in the *granularity domain* can be summed to give the original image. The method is thus a lossless transform. In [2] there is a proof that sieves preserve scale-space causality. Here, granules are denoted by  $G_s$  where  $s$  is the granule scale (area).



**Fig. 3.** Panel (A) Sieve tree computed using a morphological  $\mathcal{M}$  operator over a computer generated image. Small extremal regions (the eyes and mouth) form granules ( $G_{s_1}^1, G_{s_2}^2, G_{s_3}^3$ ) and merge to their most similar neighbour (the face) to form a new node that includes them all. The root node represents the whole image. Panel (B) shows the result of blurring the image. Nested sets of granules associated with each feature appear as near-vertical strings of nodes. For example, the open eye forms a nested set  $[G_{s=1}^2, \dots, G_{s=S}^2]$  where  $S$  is the largest granule in the set. The red node is the MSER.

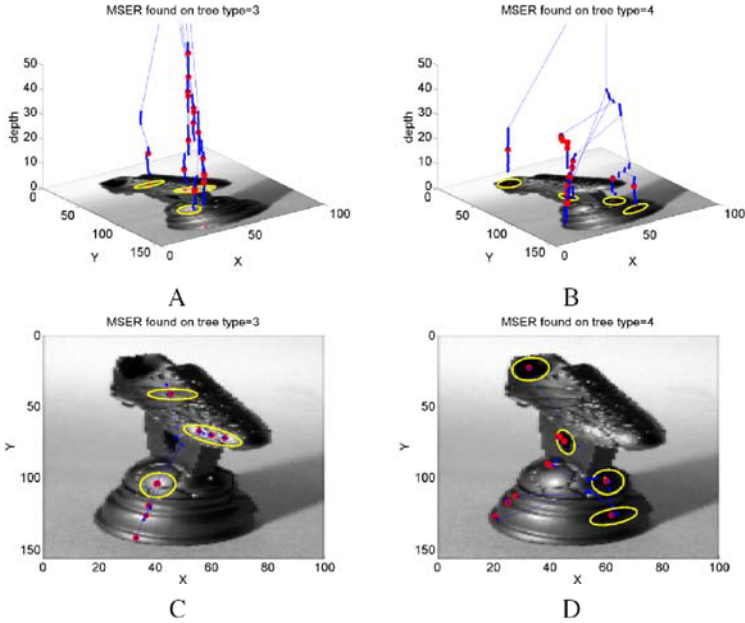
Since smaller granules are contained within larger ones, the method generates a tree [10] as illustrated in Figure 3 Panel A. Where there is a close relation between successive granules, we assign them a superscripted label, e.g. in Figure 3 Panel B. For real scene images, sievetrees may become extremely complicated and subsequent processing can be unwieldy. However, it is practical to process information associated with each node on-the-fly. For example, features can be generated such as the node area, mean colour, colour histogram moments, texture histogram moments, etc. These are described later.

### 3 Stable Regions

Maximally Stable Extremal Regions (MSER's) are known to be amongst the best performing correspondence region detectors in terms of repeatability. They are obtained by connected set openings (resp. closings) of the image as shown in Figure 3B. During the sieving process, the area of each nested granule set is monitored and the difference of areas between nodes of the same set are evaluated via equation 3. For the  $i$ th node,

$$q(i) = \frac{|Q_{i+\delta} \setminus Q_{i-\delta}|}{|Q_i|} \quad (3)$$

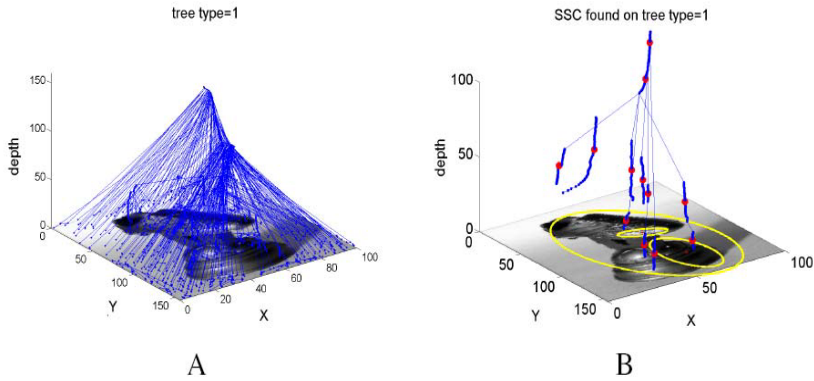
where  $Q_i$  is the  $i$ th connected set in a sequence of nested extremal regions:  $Q_1 \subset Q_2 \subset Q_3 \dots Q_i \subset \dots, | \cdot |$  denotes the number of pixels in the set and  $\delta$  is a



**Fig. 4.** An image,  $x, y$  plane, with chains of nodes that represent MSER’s plotted according to their depth,  $d$  (vertical) and mean position. (A,C) MSER’s arising from ‘openings’, or bright regions.  $B_c(x, y)$ . (B,D) Nodes arising from ‘closings’ or dark regions.  $B_o(x, y)$ . Nodes representing Maxima Stable Extremal Regions have been coloured to red whereas all other nodes from stable branches are represented with blue.

parameter of the method. The idea is to examine the sequence of  $q(i)$  to retain nodes that have local minimum  $q(i)$ . such regions are called maximally stable extremal regions. Examples of MSER’s are shown in Figure 4A and B together with a few, sample, ellipses highlighting some of the stable regions found. The algorithm, therefore, tracks locally extreme regions through increasing scale-space and the instabilities that terminate the chain result when very different neighbouring regions merge, i.e. so called branch points [6,4,13]. Continuing the closing algorithm produces a tree,  $B_c^{all}(x, y)$  where  $B_c(x, y) \subset B_c^{all}(x, y)$ . Figure 4C and D show both small and large scale MSER’s, however, large scale MSER’s tend to be unstable over different views of the same region and are often ignored.

Stable Salient Contours (SSC’s) are not very different from MSER’s. A similarity measure is defined to determine the difference between a node and its parent. In [5] Kolmogorov-Smirnoff (K-S test) was used to test for the significance of the difference between parent and child gray-level histograms. As a result of this, a score  $\lambda$  is assigned to each parent-child relationship and then, the algorithm searches exhaustively for chains of nodes where  $\lambda$  is below a threshold. If such paths are longer than a fixed value  $l$ , then that branch is designed as stable. Finally the middle node of that chain is marked as a stable node.



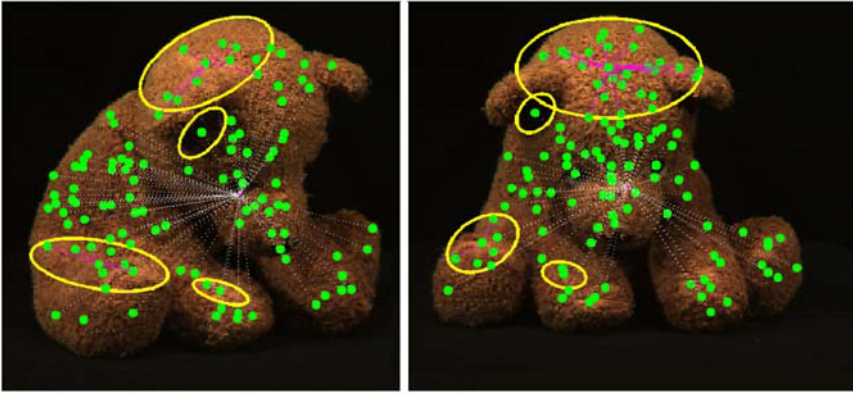
**Fig. 5.** (A) The full tree,  $B_c^{all}(x, y)$ , generated by connected set ‘closing’ over all scales,  $B_c(x, y) \subset B_c^{all}(x, y)$ . (B) SSC as described by Y.Lan where nodes representing Stable Salient Regions have been coloured to red whereas all other nodes from stable branches are represented with blue. The nodes at the middle of each stable branch are SSCs.

Perhaps the main difference between SSC’s and MSER’s is that only openings or closings trees are used to define MSER’s whereas SSC’s could be also defined over alternated sequential ( $M$  and  $N$ ) sieve trees that have been reported to be more stable [11] than simply open/close trees. As a consequence of this, SSC’s remain stable at larger scales than MSER’s. This is perhaps why SSC’s generally obtain a larger amount of correspondances than MSER’s. Figure 5 is an example of SSC’s over the same image used to extract the MSER’s defined on Figure 4.

## 4 Implementation

Our proposed implementation is capable to detect SSC’s whilst the sieve tree is being assembled. Just before the merging process takes place, when all the nearest neighbours of a extremal region are known, the Euclidean distance of colour and texture moments is computed to compare the difference between the extrema region and the new region that will be created. If this value is bellow a threshold, a path length counter will be incremented for that branch, otherwise, the stable chain is broken and the path length counter is retrieved and evaluated, if it is longer than a fixed value, then the middle node from that branch is labeled as stable. This process allows the labeling of Stable nodes ”on the flight” thus, eliminating the need for a second parse of the whole tree data structure after the sieve algorithm finishes.

One of the most important advantages of this implementation is that the data structure provides the possibility to collect different region descriptors on parallel to the merging process. We use descriptors that are based on statistical moments obtained from feature distributions (e.g. from the colour histogram for a given node). It is known that any moment about the mean, can be derived



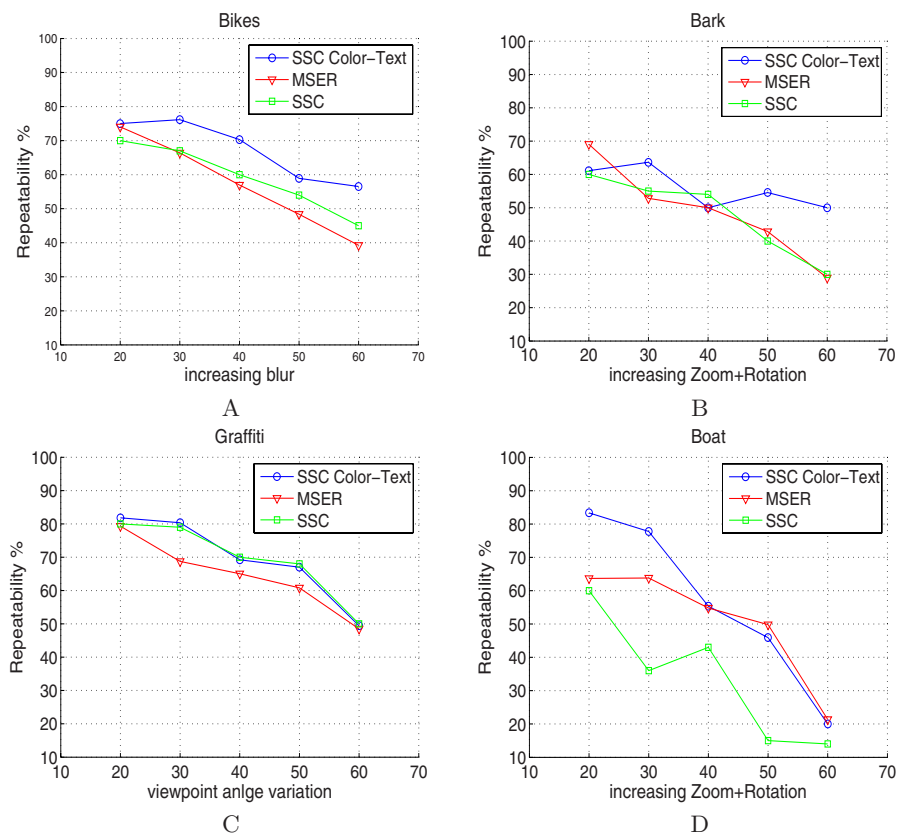
**Fig. 6.** Correspondence points found on a textured object over two different camera angle variations. Green dots represent the gravity center of SSCs obtained using colour and texture descriptors. For the sake of clarity, only a few sampled ellipses with the same second order moments of the regions that where matched on both images have been drawn.

from a combination of moments about zero which are basically sums of values at different orders.

The  $n$ th moment (about zero) of a probability density function  $f(x)$  is the expected value of  $\sum x^n$ . In our implementation we use  $\sum x^1$  (also known as  $m$ ),  $\sum x^2$  (also known as  $\mu'_2$ ) and  $\sum x^3$  (also known as  $\mu'_3$ ). Three list of size  $N$ , where  $N$  represents the total number of nodes on the image, are allocated to store the previous sums for every single node of the three. From these quantities, the calculation of the moments about the mean becomes stright forward and preserves the linearity of the algorithm as described in the following relationships:  $\mu = m$ ,  $\mu_2 = \mu'_2 - m^2$ ,  $\mu_3 = \mu'_3 - 3m\mu'_2 + 2m^3$ , where,  $\mu$ ,  $\mu_2$  and  $\mu_3$  represent the first three moments of the probability density function. Such moments are strongly related to the mean, standard deviation and the skewness and we use them to describe the probability distributions of colour and texture descriptors at each granule represented in the tree. These are used to compute the parent-child euclidean distance described previously on this section. Figure 6, illustrates Stable Salient Regions found using the proposed method.

## 5 Results

Using the standard evaluation method defined by Mikolajczyk et al [9], repeatability rates have been calculated for MSER's, conventional SSC's and the proposed alternative SSC's obtained from colour and texture moments. A set of four images (bikes, bark, graffiti and boats) are tested under different varying conditions. Each group of images consists on 6 pictures where a specific transformation is increasingly applied. Bark Images and Boat Images are both zoomed and rotated increasingly whereas a set of Bike Images are trasformed using a



**Fig. 7.** Repeatability results of four sets of images (bikes, bark, graffiti and boat) where different increasing transformations have been applied. The proposed method (SSC Color-Text) is compared against conventional SSCs and MSERs.

blurring filter. A further set of images contains pictures of the same graffiti over increasing viewpoint angle variations. In [9], homographies that map the reference frame with every image within the same group are also provided as groundtruth. Each detected region is represented as an ellipse and the test looks for its corresponding ellipse in the other images. If the overlap error is less than 40%, a correspondence is achieved. Finally, the repeatability of a region detector for a given pair of images is computed as the number of ellipse correspondences divided by the smaller of the numbers of regions generated from each image.

Figure 7 shows repeatability rates for the images in this test under varying transformations such as blur, scale, viewpoint angle, zoom and rotation. SSC's obtained from colour and texture moments (SSC Color-Text) show improvements on the bikes (panel A) and Bark images (panel B) compared to the other methods and it is at least as good as conventional SSC's on the graffiti images (panel C) and relatively comparable to MSER's for the Boat images. For this image data

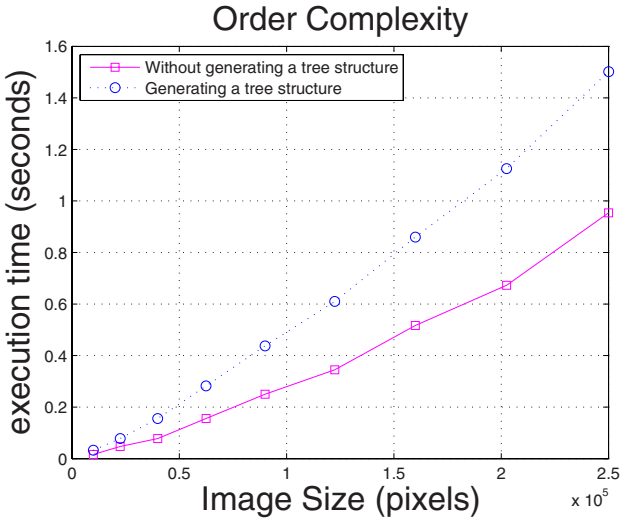


Fig. 8. Complexity of Structured Images

set, our method seems to produce generally better results than those produced by these two top performing approaches with a further advantage of the efficiency and speed of our implementation. The order complexity of the algorithm is nearly linear as shown in figure 8. It takes approximately 1.5 seconds to process a 400x600 image on a Pentium IV. This improves dramatically the efficiency of conventional SSC's whilst, at the same time, achieves better repeatability rates than the other two methods.

## 6 Conclusions

It has been shown here how Stable Salient Contours (SSC's) and Maximally Stable Extremal Regions (MSERs) may be generated using sieve trees. A new implementation of the sieve algorithm that uses color and texture moments to detect and describe SSC's has been introduced and evaluated against MSER's and conventional SSC's. Results show improvements on repeatability rates against both methods. The proposed Structured Images not only generate MSER's and conventional SSC's but can also generate alternative SSC's obtained from colour and texture moments that are capable of deliver better stable regions than its predecessor under a reasonable amount of time. Finally, the complexity of the algorithm was found to be nearly linear as opposite to conventional SSC's. Nevertheless MSER's, remain the fastest method for stable region detection but, as described in section 2, it becomes unstable at larger scales and therefore it achieves lesser correspondance rates than those obtained via SSC's.



## References

1. Bangham, J., Hidalgo, J., Harvey, R., Cawley, G.: The segmentation of images via scale-space trees (1998)
2. Bangham, J.A., Chardaire, P., Ling, P., Pye, C.J.: Multiscale nonlinear decomposition: the sieve decomposition theorem. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18, 518–527 (1996)
3. Gibson, S., Harvey, R.: Trecognition and retrieval via histogram trees. In: *British Machine Vision Conference*, vol. 2, pp. 531–540 (2001)
4. Koenderink, J.J.: The structure of images. *Biological Cybernetics* 50, 363 (1984)
5. Lan, Y., Harvey, R., Perez-Torres, R.: Finding stable salient contours. In: *Proceedings of the British Machine Vision Conference*, Oxford (2005)
6. Bretzner, L., Lindeberg, T.: On the handling of spatial and temporal scales in feature tracking. In: *Proc. First Int. Conf. on Scale-space theory*, pp. 128–139. Springer, Heidelberg (1997)
7. Lowe, D.: Sift: Distinctive image features from scale invariant keypoints. In: *Proceedings of IJCV*, vol. 1, pp. 91–110 (2004)
8. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: *Proceedings of the British Machine Vision Conference*, London, vol. 1, pp. 384–393 (2002)
9. Tuytelaars, T., Schmid, C., Zisser-Man, A., Matas, J., Schaffalitzky, F., Kadir, T., Mikolajczyk, K., van Gool, L.: A comparison of affine region detectors. *International Journal of Computer Vision*, 43–72 (2005)
10. Harvey Moravec, R., Bangham, J.A.: Scale trees for stereo vision. *IEE Proceedings: Vision, Image and Signal Processing* 147(4), 363–370 (2000)
11. Bosson, A., Harvey, R., Bangham, J.A.: Robustness of some scale-spaces. In: *British Machine Vision Conference (BMVC 1997)*, Colchester, UK, vol. 1, pp. 11–20 (1997)
12. Southam, P.: Texture granularities. In: *International Conference on Image Analysis and Processing*, Italy, vol. 1, pp. 233–240 (2005)
13. Witkin, A.P.: Scale-space filtering. In: *8th International Joint Conference on Artificial Intelligence*, pp. 1019–1022 (1983)

# The Wavelet Based Contourlet Transform and Its Application to Feature Preserving Image Coding

Osslan Osiris Vergara Villegas<sup>1</sup> and Vianey Guadalupe Cruz Sánchez<sup>2</sup>

<sup>1</sup> Universidad Autónoma de Ciudad Juárez (*UACJ*)  
Avenida del Charro No. 450 Norte, Ciudad Juárez Chihuahua México  
overgara@uacj.mx

<sup>2</sup> Centro Nacional de Investigación y Desarrollo Tecnológico (*cenidet*)  
Interior Internado Palmira S/N. Col. Palmira, Cuernavaca Morelos México  
vianey@cenidet.edu.mx

**Abstract.** The Contourlet Transform (CT) can capture the intrinsic geometrical structure of an image. The CT is a redundant transform, and for coding applications this can be a disadvantage. In order to avoid the contourlet redundancy we change the pyramidal stage by a multiscale stage. The new non redundant transform is called: The Wavelet Based Contourlet Transform. We take the advantages offered by the new transform to build a novel feature preserving image coder. The preservation is made both: by a stage of feature definition and extraction and by a proposed modified version of the SPIHT coder. SPIHT modification allows selecting a transform coefficient not only by magnitude, but as pertaining to a feature of interest map. We present tests in order to demonstrate the good performance of the coder. Finally, we compare the results with other existing methods. The coder designed performs well even at very low bit rates.

## 1 Introduction

The last decade we have seen a great interest into designing mathematical and computational tools based on multiscale ideas. In different sciences, the development of multiscale methods led to convenient tools to navigate through large datasets, to transmit compressed data rapidly, to remove noise from signals and images, and to identify crucial transient features in such datasets [1].

The Discrete Wavelet Transform (DWT) is a tool that can be applied on the discrete data to obtain a multiscale representation of the original data. From the digital point of view, the original information must be represented and delivered in efficient form. The representation efficiency, talks about the ability to capture significant information of an object of interest in a small description. From the practical point of view this representation is obtained by means of structured transformations and fast algorithms [2].

The algorithms based on wavelets have been proved to work well for different image processing tasks including compression. The DWT is a powerful tool to detect and separate image singularities in the horizontal and the vertical directions, but if the directions are different to those mentioned the DWT does not performs very well.

To solve the problem of directionality, the Contourlet Transform (CT) was created by Do and Vetterli [3]. The CT allows for different number of directions at each scale/resolution to nearly achieve critical sampling. The CT is designed to capture high frequency components that represent directionality. Unfortunately, the CT is not adequate for coding purposes because it is a redundant transform.

Several important features of images can be lost at the quantization stage of a lossy image coder. There are some areas such as medicine in which there are laws and restrictions about the use of original images. An image can not be compressed without the information of a tumor or other cue informing about some disease of a patient. When a coder is designed it is important to ensure that in certain parts of the image, information loss do not occur, these leads to “Feature preserving image coding” [4].

In this paper, we address the creation of a novel image coder that allows the preservation of important image features such as edges and textures. There are two main stages to assure the success of the coder proposed. The first one is the use of a new transform called the Wavelet Based Contourlet Transform (WBCT), and the second one is a proposed modification for the SPIHT algorithm.

The outline of the paper is as follows: Section two gives the details of the proposed image coder, the tests and results using typical images and the WBCT are showed in section three. Finally, the conclusions and further works can be found in section four.

## 2 Feature Preserving Image Coding Methodology

Images are analyzed as a composition of: edges, textures and edge associated details [5]. In order to design a feature preserving lossy image coder, it is important to identify the image features (textures, edges) to preserve, and then we can use lower quality in other image regions. The goal of the coder is to use the decompressed images for future computer vision or pattern recognition applications.

Given an input image, we first obtain the important features (edges, textures). This is made by a feature detection process. After feature detection, we compute the WBCT in order to change the domain of the input image. When we have the features and the transformation, a process of mapping of the corresponding features positions to the new domain is made. To code an image a modified version of SPIHT algorithm is applied. Finally an arithmetic coding process is applied to the stream obtained from SPIHT. To decode an image the inverse process is made. In the following subsections we offer a brief description of each stage for feature preserving image coder.

### 2.1 Feature Detection

First, we need to detect image important information (edges and textures). For this, we use the edge detector called Smallest Univalued Segment Assimilating Nucleus (SUSAN). SUSAN is a more robust and effective method than Canny [5] in the sense that it provides much better edge localization and connectivity. SUSAN use a predetermined window centered on each image pixel, applying a locally acting set of rules to give an edge response. This response is then processed to give as output a set of edges [6]. In summary SUSAN performs the following three steps at each image

pixel: a) Place a circular mask around the pixel in question (the nucleus), b) Calculate the number of pixels within the circular mask which have similar brightness to the nucleus, and c) Subtract the USAN size from the geometric threshold to produce an edge strength image.

Once we detect the edges of an image we know, that the rest of the information not pertaining to edges can be considered as background textures. The map of textures or edges is used for the stage of pixel mapping.

## 2.2 The Wavelet Based Contourlet Transform

The CT is a redundant transform, and this represents a disadvantage for image coding. The redundancy of the CT occurs at the pyramidal stage. As a result of pyramidal filtering we obtain two images, the first one resulting from low pass approximation, and the second one obtained from the high pass approximation.

The image of details obtained (resulting from high pass) has always the same size of the immediately anterior; this is because there are not image resolution reductions. The directional decomposition is computed with the detailed image, by that, if we made more pyramidal decompositions we generate at least a half more information of the previous level as redundancies.

To take advantage of the directionality offered by CT and to avoid the redundancy, we can change the pyramidal filter by a wavelet filter. This is the main idea behind a new non redundant transform called Wavelet Based Contourlet Transform (WBCT) [7]. In order to perform the WBCT it is important to ensure that we can obtain the perfect reconstruction of an image for the best case. The WBCT is as follows:

### 1. Compute the DWT of an image:

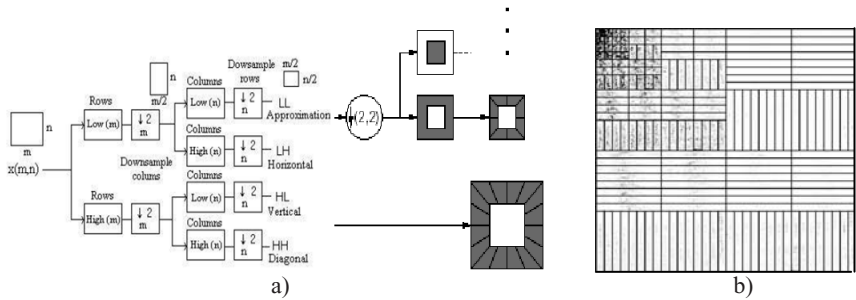
- 1.1. Select and design the decomposition filters (low pass and high pass) corresponding to the wavelet family used. For this case the Biorthogonal 2.2.
- 1.2. Select the image extension; this is to compute the image convolution. As a result we obtain an extended image ( $Imext$ ). We use the image periodic extension.
- 1.3. Convolve the rows and columns of  $Imext$  with the low pass filter. Then, perform image downsampling to obtain an approximation image ( $LL$ ).
- 1.4. Convolve  $Imext$  rows with the low pass and columns with the high pass. Then, perform image downsampling to obtain the horizontal coefficients ( $LH$ ).
- 1.5. Convolve  $Imext$  rows with the high pass and columns with the low pass. Then, perform image downsampling to obtain the vertical coefficients ( $HL$ ).
- 1.6. Convolve  $Imext$  rows and columns with the high pass filter. Then, perform image downsampling to obtain the diagonal coefficients ( $HH$ ).
- 1.7. Repeat steps 1.2 – 1.6 using the  $LL$  image until the defined decomposition level is reach. For this paper, we select five levels.

### 2. Design the directional filters, for this paper we use the directional PKVA filters.

3. Performs the directional decomposition using the images:  $LH$ ,  $HL$  and  $HH$ . The process is made using a bidimensional filter bank that decomposes an image with a maximum of 5 directions or 32 subbands at the finer wavelet subband.

4. Repeat the step three with the next level of *LH*, *HL* and *HH* images by decreasing the direction until reach two directions or four subbands. The final approximation image of the WBCT is a pure wavelet; we not compute the directional decomposition.

In figure 1 we show the process to compute the WBCT, first an image is wavelet decomposed and then the transform is used as an input of the contourlet directional stage. As you can observe from figure 1b the WBCT is not redundant.



**Fig. 1.** The Wavelet Based Contourlet Transform. a) The process to obtain the WBCT and b) Example of the WBCT for the camman image.

### 2.3 Pixel Mapping

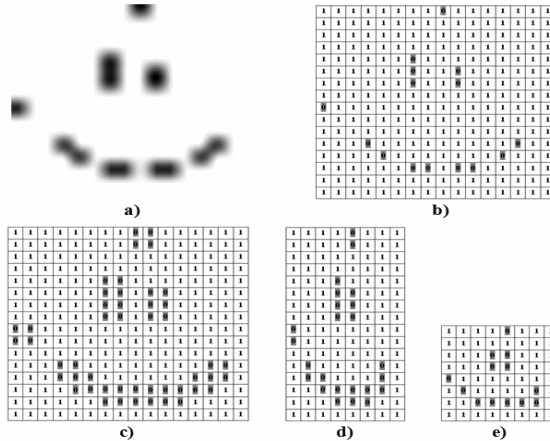
The points obtained with SUSAN are used to design the feature map at the transform domain. We know that at the transform domain a coefficient of scale *i* have an area of  $2i \times 2i$  positions of the original domain; there exist a hierarchic relation between the coefficients which allows defining a spatial orientation tree.

Using the hierarchic relationship we can build the corresponding map of the edges or of the textures obtained at the original domain but now at the transform domain.

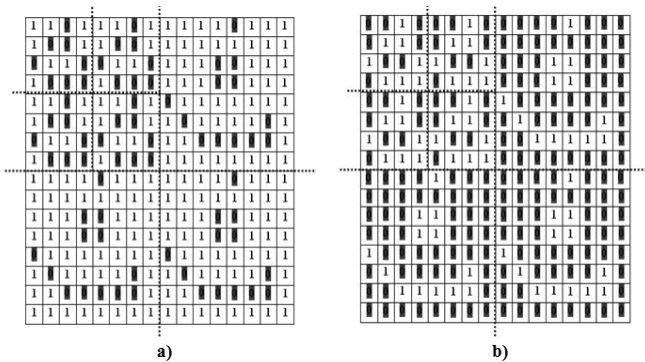
To explain the pixel mapping process we use the images of figure 2. First, we detect the edges of the input image (2a) obtaining the image of figure 2b. Second, we perform the upsampling to the image rows and columns to obtain figure 2c. Third, we perform the downsampling to the image columns (figure 2d). Finally, we perform the downsampling to the image rows (figure 2e).

The image 2e is used to build the finer subbands of the transform domain. The process is repeated until the number of decomposition levels is reached and the complete map is made as it is show in figure 3a. Finally, if we need to preserve textures the map obtained at transform domain is inverted as it is shown in figure 3b.

If we need to preserve both edges and textures we can use the complete map. At the end of the process we always change to one the first four pixel of the left superior square in order to avoid errors at the SPIHT coding stage.



**Fig. 2.** Pixel mapping process. a) Smiley face, b) Edge map, c) Upsampled image, d) Image 2c columns downsampling and e) Rows of the image 2d downsampling.



**Fig. 3.** Transform domain map obtained with pixel mapping. a) Edge map and b) Texture map.

**2.4 Image Coding with Modified SPIHT**

In 1996 Said and Pearlman present an improved version of the Embedded Zerotree Wavelet (EZW) coder. The authors propose a different tree structure called Set Partitioning In Hierarchical Trees (SPIHT) [8]. The principle behind SPIHT is to define significance of a pixel if its value is larger or equal to given threshold.

The SPIHT modification proposed here allows the definition of a significance pixel both by significance and by the pixel position corresponding to the pixel mapping in the WBCT domain. A similar idea was proposed in [4] and the main differences are: we use SPIHT instead of EZW, in our model we can preserve two or more features at same time, and we use a different wavelet filter.

In order to illustrate how SPIHT modification works we use table 1 and 2 for SPIHT and for the modification. We use the figure 4 in order to give an example. In figure 4a a portion of a 4 x 4 image is shown and in figure 4b the edge map is shown.

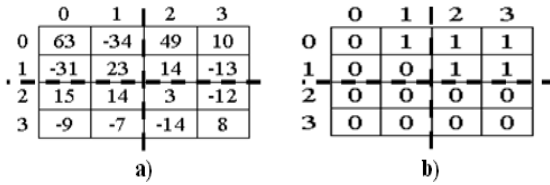


Fig. 4. Image portions. a) 4 x 4 portion of an image and b) WBCT edge map.

First  $O(i, j)$  is defined as the set of offspring (direct descendants) of a tree node defined by pixel location  $(i, j)$ .  $D(i, j)$  is the set of descendants of a node defined by pixel location  $(i, j)$ .  $L(i, j)$  is the set defined by  $L(i, j) = D(i, j) - O(i, j)$ . The following explanation refer to the numbered entries in table 1 for proposed modified SPIHT and compared to original SPIHT shown in table 2.

1. Initial settings. The threshold is set to 32, and  $LIS$  (List of Insignificant Sets),  $LIP$  (List of Insignificant Pixels) and  $LSP$  (List of Significant Pixels) are initialized.
2. SPIHT begins coding the significance of the  $LIP$  pixels. The position  $(0, 0)$  is insignificant and  $(0, 1)$  is significant because is larger than the threshold and it is part of the map. Different to original (table 2) in which the two coordinates  $(0, 0)$  and  $(0, 1)$  are significant. For both schemes the positions  $(1, 0)$  and  $(1, 1)$  are insignificant.
3. After testing pixels, SPIHT begins to test sets, following the  $LIS$  entries.  $D(0, 1)$  is the set of four coefficients  $\{(0, 2), (0, 3), (1, 2), (1, 3)\}$ . Because  $D(0, 1)$  is significant SPIHT test the significance of the four offsprings. Finally  $(0, 1)$  is removed from  $LIS$ .
4. Same procedure to the comment 3 is applied with  $D(1, 0)$ , since is insignificant no action need to be taken, and check for the next element of  $LIS$ .
5.  $D(1, 1)$  is insignificant no action need to be taken, the first pass ends and the refinement pass starts and is made equals to the original SPIHT algorithm.

Table 1. Coding with modified SPIHT

Comment	Pixel or set tested	Output bit	Action	Control list
(1)				$LIS = \{(0,1)A, (1,0)A, (1,1)A\}$ $LIP = \{(0,0), (0,1), (1,0), (1,1)\}$ $LSP = \phi$
(2)	$(0,0)$	0	none	
	$(0,1)$	1-	$(0,1)$ to LSP	$LIP = \{(0,0), (1,0), (1,1)\}$ $LSP = \{(0,1)\}$
	$(1,0)$	0	none	
	$(1,1)$	0	none	
(3)	$D(0,1)$	1	Test offsprings	$LIS = \{(0,1)A, (1,0)A, (1,1)A\}$
	$(0,2)$	1+	$(0,2)$ to LSP	$LSP = \{(0,1), (0,2)\}$
	$(0,3)$	0	$(0,3)$ to LIP	$LIP = \{(0,0), (1,0), (1,1), (0,3)\}$
	$(1,2)$	0	$(1,2)$ to LIP	$LIP = \{(0,0), (1,0), (1,1), (0,3), (1,2)\}$
	$(1,3)$	0	$(1,3)$ to LIP	$LIP = \{(0,0), (1,0), (1,1), (0,3), (1,2), (1,3)\}$
				$LIS = \{(1,0)A, (1,1)A\}$
(4)	$D(1,0)$	0	none	
(5)	$D(1,1)$	0	none	

Table 2. Coding with original SPIHT

Comment	Pixel or set tested	Output bit	Action	Control list
(1)				$LIS = \{(0,1)A, (1,0)A, (1,1)A\}$ $LIP = \{(0,0), (0,1), (1,0), (1,1)\}$ $LSP = \phi$
(2)	$(0,0)$	1+	$(0,0)$ to LSP	$LIP = \{(0,1), (1,0), (1,1)\}$ $LSP = \{(0,0)\}$
	$(0,1)$	1-	$(0,1)$ to LSP	$LIP = \{(1,0), (1,1)\}$ $LSP = \{(0,0), (0,1)\}$
	$(1,0)$	0	none	
	$(1,1)$	0	none	
(3)	$D(0,1)$	1	Test offsprings	$LIS = \{(0,1)A, (1,0)A, (1,1)A\}$
	$(0,2)$	1+	$(0,2)$ to LSP	$LSP = \{(0,0), (0,1), (0,2)\}$
	$(0,3)$	0	$(0,3)$ to LIP	$LIP = \{(1,0), (1,1), (0,3)\}$
	$(1,2)$	0	$(1,2)$ to LIP	$LIP = \{(1,0), (1,1), (0,3), (1,2)\}$
	$(1,3)$	0	$(1,3)$ to LIP	$LIP = \{(1,0), (1,1), (0,3), (1,2), (1,3)\}$
				$LIS = \{(1,0)A, (1,1)A\}$
(4)	$D(1,0)$	0	none	
(5)	$D(1,1)$	0	none	

Using a bit rate of 5 for figure 3a, then the list with original SPIHT is:  $LSP = \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (0, 3), (1, 2), (1, 3), (2, 0), (2, 1), (3, 0), (2, 3), (3, 2),$

$(3, 3), (3, 1)$  and the bits spend are 55. With modified SPIHT  $LSP = \{(0, 1), (0, 2), (0, 3), (1, 2), (1, 3)\}$ , corresponding to the positions to preserve and the bits spend are 56.

Finally the bit stream obtained from modified SPIHT is entropy coded to obtain the final compressed file, to the stream we add information about image size, wavelet level and filter used and directional level and filter used. To decompress an image we perform all the steps at inverse mode.

### 3 Simulation, Results and Comparisons

We perform the feature preserving image coding experiments using the WBCT to preserve: edges, textures and edges and textures in a joint way. The experiments were conducted on many classical images. We use two kind of images RGB (Lena, Barbara and Baboon) and grayscale (Clown, Buthfish and Camman) shown in figure 5.

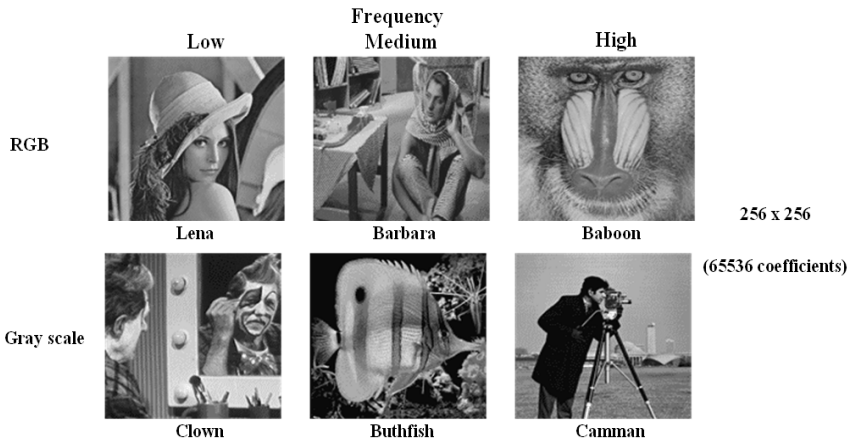


Fig. 5. Images used for the tests of feature preserving image coding

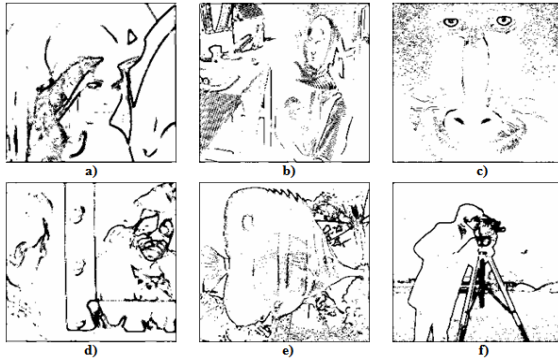
The correspondent edge maps for each image are shown in figure 6. Table 3 shows the threshold to obtain edge maps and the number of points pertaining to edges. We know that all the point not corresponding to edges are points pertaining to texture cue.

Table 3. Thresholds and edge points obtained with SUSAN

<i>Image</i>	<i>Threshold</i>	<i>Number of points:</i>		
		<i>R</i>	<i>G</i>	<i>B</i>
<i>Lena</i>	50	8036	7440	6087
<i>Barbara</i>	73	8230	6132	6351
<i>Baboon</i>	65	4597	7515	7688
<i>Clown</i>	58	7085	0	0
<i>Buthfish</i>	70	7050	0	0
<i>Camman</i>	50	7505	0	0



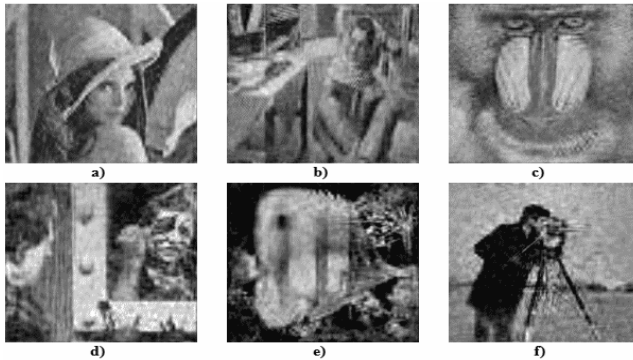
The tests are made in a subjective and objective way, we compute the Compression Factor ( $C. F.$ ), four objective measures: Mean Square Error ( $MSE$ ), Peak Signal to Noise Ratio ( $PSNR$ ), Frobenius Norm ( $F$ ) and Norm two ( $N_2$ ), and one subjective measure: Picture Quality Scale ( $PQS$ ). For the final test we add the measure Mean Opinion score ( $MOS$ ), obtained by six observers and measured in a scale from 1 to 5.



**Fig. 6.** Edge maps. a) Lena, b) Barbara, c) Baboon, d) Clown, e) Buthfish and f) Camman.

### 3.1 Test One: Edge Preserving Image Coding

The first test is for edge preserving. The compression is made with a bit rate of 0.1 to obtain a compression factor of 80: 1. In figure 6 we show the decompressed images for this test, and the error measures are shown in table 4.



**Fig. 7.** Edge preserving. a) Lena, b) Barbara, c) Baboon, d) Clown, e) Buthfish and f) Camman.

From figure 7, the edge preserving is made in a good way even at very low bit rates. We can use the camman to made comparison with another existing literature methods. The work of [9] obtain a  $PSNR = 22.39$ , with the work of [10] the  $PSNR = 22.39$ , finally with the proposed model we obtain a  $PSNR = 20.60$ . The objective measures can not give information about the goodness of the edge preserving, even

with these measures the proposed model visually performs well than the other methods because more details were preserved. The images obtained by [9] and [10] have the biggest PSNR, but in the image of [9] a great part of the camera tripod, details of the face and the background are lost. With the image obtained by [10] the same situation occurs.

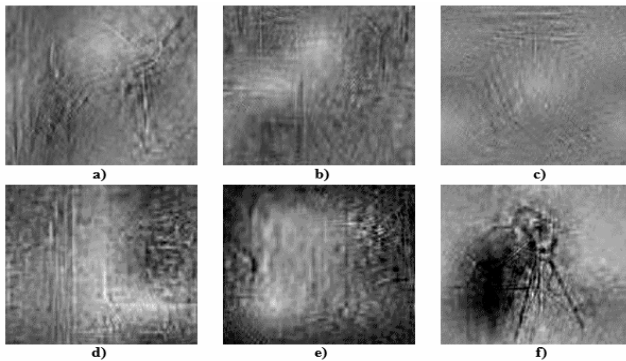
**Table 4.** Error measures obtained for test 1

<i>Image</i>	<i>C. F.</i>	<i>MSE</i>	<i>PSNR</i>	<i>F</i>	<i>N<sub>2</sub></i>	<i>PQS</i>
<i>Lena</i>	82.643:1	301.3627	23.4189	0.1347	0.0356	-2.0078
<i>Barbara</i>	81.647:1	652.7386	19.9833	0.2102	0.0502	-0.7390
<i>Baboon</i>	83.343:1	537.0047	20.8431	0.1705	0.0431	-0.4649
<i>Clown</i>	80.610:1	428.2685	21.8136	0.15212	0.0423	-2.8973
<i>Buthfish</i>	82.539:1	899.213	18.5922	0.2608	0.0652	-0.2627
<i>Camman</i>	84.236:1	565.1704	20.609	0.1774	0.0601	-4.3694

Immediate observation from table four suggests that the better reconstructed image is Lena. The poorest reconstructed image is Buthfish. From the compression factor we can obtain a little additional compression by the stage of arithmetic coding.

### 3.2 Test Two: Texture Preserving Image Coding

The second test is to verify the texture preservation capabilities of the coder. For this, we use the edge maps, and select the values not corresponding to edges in order to obtain the texture map. Again the tests were made with a bit rate of 0.1. The decompressed images are shown in figure 8, and the error measures are shown in table 5.



**Fig. 8.** Texture preserving. a) Lena, b) Barbara, c) Baboon, d) Clown, e) Buthfish, f) Camman.

Generally, the human beings analyze an image based on the information offered by the edges. But, if you observe the image carefully you can recognize the objects of the images even without the edge information. Talking about the compression factor, we can obtain approximately a twenty percent of additional compression without

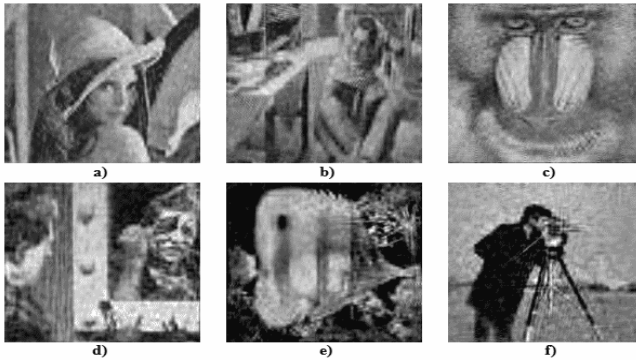
**Table 5.** Error measures obtained for test 2

<i>Image</i>	<i>C. F</i>	<i>MSE</i>	<i>PSNR</i>	<i>F</i>	<i>N<sub>2</sub></i>	<i>PQS</i>
<i>Lena</i>	104.190:1	1574.4258	16.453	0.2997	0.15711	-8.2182
<i>Barbara</i>	104.025:1	1934.6475	15.2795	0.3628	0.1535	-4.5879
<i>Baboon</i>	107.260:1	1951.3075	15.4217	0.3236	0.2121	-4.0749
<i>Clown</i>	98.847:1	2834.8954	13.6054	0.3913	0.2324	-9.9763
<i>Buthfish</i>	107.789:1	3054.3474	13.2816	0.4807	0.244	-6.7532
<i>Camman</i>	97.523:1	1610.0132	16.0625	0.2994	0.1539	-9.1957

information lost. This is due texture uniformity, which is reflected in the bit stream obtained with SPIHT and coded more compactly with the arithmetic coding stage. By analyzing images individually, the best decompressed image is Camman. This is due because Camman is a high frequency image and does not have much texture information. The poorest decompressed image is Buthfish.

### 3.3 Test Three: Feature Preserving Image Coding

The final test is made for texture an edge preserving in a joint way, with a bit rate of 0.1. The decompressed images are shown in figure 9. The error measures obtained for test three are shown in table 6.

**Fig. 9.** Feature preserving. a) Lena, b) Barbara, c) Baboon, d) Clown, e) Buthfish, f) Camman.**Table 6.** Error measures obtained for test 3

<i>Image</i>	<i>C. F.</i>	<i>MSE</i>	<i>PSNR</i>	<i>F</i>	<i>N<sub>2</sub></i>	<i>MOS</i>	<i>PQS</i>
<i>Lena</i>	82.262:1	286.3876	23.6843	0.1310	0.0291	2.1	-1.8934
<i>Barbara</i>	81.276:1	610.7459	20.2781	0.2033	0.0436	2.2	-0.5912
<i>Baboon</i>	82.852:1	506.4736	21.1482	0.1655	0.0299	3.1	-0.3112
<i>Clown</i>	80.412:1	400.952	22.0999	0.1471	0.0408	2.3	-2.6304
<i>Buthfish</i>	81.613:1	830.7673	18.936	0.2507	0.0540	2.8	0.3315
<i>Camman</i>	82.957:1	454.7946	21.5527	0.1591	0.0378	1.5	-3.8443

As a result of this test we obtain the better decompressed images, the ringing and the granularity presented in images is because we not preserve the other information called edge associated details. For the three tests the compression factors are good to compare with other similar algorithms adding the advantage that we do not send any type of side information.

## 4 Conclusions

In this paper we present a novel feature preserving image coder. The main contribution of the methodology presented is not in the reduction of the errors measurement between the original and decompressed images. The main contribution is in the correct preservation and reconstruction of the important features of an image such as edges and textures even at very low bit rates.

The possibility of coding images with feature preserving offers a wide range of applications in several industries in which this process is imperative such as medicine, mobile devices and face recognition systems.

In the future we are going to work to avoid the granularity appeared in the images due to pixel mapping process.

## References

1. Candès, E.J., Demanet, L., Donoho, D., Ying, L.: Fast Discrete Curvelet Transforms, Technical Report: Applied and Computational Mathematics, California Institute of Technology (2005)
2. Do, M.N.: Directional Multiresolution Image Representations, Ph. D. thesis, Signal Processing Laboratory, Lausanne Federal Polytechnic School (EPFL), Lausanne, Swiss (October 2003)
3. Do, M.N., Vetterli, M.: The Contourlet Transform: An Efficient Directional Multiresolution Image Representation. *IEEE Transactions on Image Processing* 14(12), 2091–2106 (2005)
4. Namuduri, K.R., Ramaswamy, V.N.: Feature Preserving Image Compression. *Pattern Recognition Letters* 24(15), 2767–2776 (2003)
5. Canny, J.F.: A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8(6), 679–698 (1986)
6. Smith, S.M., Brady, J.M.: SUSAN - A New Approach to Low Level Image Processing. *International Journal of Computer Vision* 23(1), 45–78 (1997)
7. Eslami, R., Radha, H.: Wavelet-Based Contourlet Transform and its Application to Image Coding. In: *Proceedings of the International Conference on Image Processing (ICIP)*, vol. 5, pp. 3189–3192 (2004)
8. Said, A., Pearlman, W.A.: A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees. *IEEE Transactions on Circuits and Systems for Video Technology* 6, 243–250 (1996)
9. Mertins, A.: Image Compression Via Edge-Based Wavelet Transform. *Optical Engineering* 38(6), 991–1000 (1999)
10. Schilling, D., Cosman, P.: Feature-Preserving Image Coding for Very Low Bit Rates. In: *Proceedings of the IEEE Data Compression Conference (DCC)*, pp. 103–112 (2001)

# Design of an Evolutionary Codebook Based on Morphological Associative Memories

Enrique Guzmán<sup>1</sup>, Oleksiy Pogrebnyak<sup>2</sup>, and Cornelio Yañez<sup>2</sup>

<sup>1</sup> Universidad Tecnológica de la Mixteca  
eguzman@mixteco.utm.mx

<sup>2</sup> Centro de Investigación en Computación del Instituto Politécnico Nacional  
(olek, cyanez)@pollux.cic.ipn.mx

**Abstract.** A central issue in the use of vector quantization (VQ) for speech or image compression is the specification of the codebook. In this paper, the design of an evolutionary codebook based on morphological associative memories (MAM) is presented. The algorithm proposed for codebook generation involves two steps. First, having a set of images, one of the images is chosen to create the initial codebook. The algorithm applied to the image for codebook generation uses the morphological autoassociative memories (MAAM). Second, an evolution process of codebook creation occurs applying the algorithm on new images. This process adds the information codified of the next image to the codebook allowing to recover the images with better quality without affecting the processing speed. The performance of the generated codebook is analyzed in case when MAAM in both *max* and *min* categories are used. The presented algorithm was applied to image set after discrete cosine transformation followed by a quantization process. The proposed algorithm has a high processing speed and provides a notable improvement in signal to noise ratio.

**Keywords:** morphological associative memories, codebook generation, evolutionary codebook.

## 1 Introduction

Vector quantization (VQ) is a technique that can produce results very next to the theoretical limits. Its main disadvantage is that it is a very complex process. VQ can be divided on two sub-processes: codebook generation and codeword search process. For the codebook generation, techniques of high complexity are used, and the process of search for suitable reconstruction vector for the given input vector is very slow.

This work focuses in the codebook generation process. A great variety of algorithms developed in this area is known. The simplest scheme to find a codebook is the LBG algorithm by Y. Linde, A. Buzo and R. Gray [1], [2], which is a practical suboptimal clustering analysis algorithm. This algorithm uses the Euclidean distance for codebook generation. LBG algorithm is a very slow process, because in each iteration an input vector is compared with each codified vector of the codebook.

W. H. Equitz in [3] proposed an algorithm known as pairwise nearest neighbor (PNN), which is also widely used in clustering analysis. The PNN algorithm derives a vector quantization codebook in a diminishingly small fraction of the time previously required, without sacrificing performance. In addition, the time needed to generate a codebook grows only like  $O(N \log N)$  in training set size, and is independent of the number of codeword desired. This method permits either to minimize the number of required codewords subject to a maximum allowable distortion or minimize the distortion subject to maximum rate.

Codebooks generated by LBG and PNN algorithms guarantee local minimum distortion, but not global minimum distortion. To solve this problem, simulated annealing algorithm (SA) was proposed by Jacques Vaisey and Allen Gersho [4], [5]. This method can improve the codebook, but it is significantly more complex. SA is a procedure that employs randomness in a search algorithm and tends to skirt relatively poor local minima in favor of better ones. By altering the LBG with the techniques motivated by SA it is possible to improve the codebook quality reducing the distortions involved in training set coding.

A clustering technique called self-organizing feature map was proposed by Kohonen as a learning algorithm for neural networks [6]. The Kohonen learning algorithm (KLA) is an “on-line” algorithm where the codebook is designed while training data input, and the reduction of the distortion function is not necessarily monotonic. C. Amerijckx et al proposed a compression scheme for digital still images, by using the Kohonen’s neural network algorithm [7], [8].

In this paper, an algorithm for evolutionary codebook generation based on Morphological Autoassociative Memories (MAAM) is proposed. Morphological associative memories (MAM) have low computational complexity since they use only sums and comparisons in their operation. MAM are known to be an excellent tool in the recognition and to recovery of patterns, even if these exhibit dilative, erosive or random noise. These features allow the use of MAAM in both codebook generation and codeword search process. The performance of the generated codebook is comparatively analyzed for cases when MAAM in both *max* and *min* categories are used and when the proposed algorithm is applied to image set after discrete cosine transform (DCT) and quantization process. The resulted algorithm has a high processing speed and improves signal to noise ratio of restored images.

## 2 Morphological Associative Memories

In 1998, Ritter, Sussner and Diaz de León created morphological associative memories [9]. The MAM base their operation on the morphological operations, dilation and erosion. In other words, they use the maximums or minimums of sums in their operation. This feature distinguishes them from the classical associative memories that use sums of products.

Let the input patterns and output patterns for the associative memory are represented by  $\mathbf{x} = [x_i]_n$  and  $\mathbf{y} = [y_i]_m$  respectively. Next, let  $\{(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), \dots, (\mathbf{x}^k, \mathbf{y}^k)\}$  be  $k$  vector pairs defined as a *fundamental set of associations*. The fundamental set of associations is represented by:

$$\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \dots, k\}. \quad (1)$$

An associative memory is represented by a matrix and is generated on the fundamental set of associations. Once the fundamental set is delineated, one can use the necessary operations for the learning process and recovery process of MAM. These operations are the *maximum product* and *minimum product* and they use the maximum  $\vee$  and minimum  $\wedge$  operators [10], [11], [12].

According to the mode of operation, the associative memories are classified in two groups: morphological autoassociative memories (MAAM) and morphological heteroassociative memories (MHM). Morphological autoassociative memories are of particular interest for the development of this work.

## 2.1 Morphological Autoassociative Memories

A MAM is autoassociative if  $\forall \mu \in \{1, 2, \dots, k\}$ , it holds  $\mathbf{x}^\mu = \mathbf{y}^\mu$ . The fundamental set of associations is defined as:  $\{(\mathbf{x}^\mu, \mathbf{x}^\mu) \mid \mu = 1, 2, \dots, k\}$ . There are two categories of MAAM: *max*, symbolized by  $\mathbf{M}$ , and *min*, symbolized by  $\mathbf{W}$ .

### 2.1.1 Morphological Autoassociative Memories *max*

The  $\mathbf{M}$  memories are those that use the minimum product and the maximum operator in their learning phase and the minimum product in their recovery phase.

#### *Learning phase:*

1. For each  $k$  element of the fundamental set of associations  $(\mathbf{x}^\mu, \mathbf{x}^\mu)$ , the matrices  $\mathbf{y}^\mu \Delta (-\mathbf{x}^\mu)^\vee$  are calculated.
2. The  $\mathbf{M}$  memory is obtained applying the maximum operator  $\vee$  to the matrices resulting from step 1.  $\mathbf{M}$  is given by

$$\mathbf{M} = \bigvee_{\mu=1}^k [\mathbf{x}^\mu \Delta (-\mathbf{x}^\mu)^\vee] = [m_{ij}]_{m \times n} \quad (2)$$

$$m_{ij} = \bigvee_{\mu=1}^k (x_i^\mu - x_j^\mu)$$

#### *Recovery phase:*

1. The minimum product  $\mathbf{M} \Delta \mathbf{x}^\omega$  is calculated, where  $\omega \in \{1, 2, \dots, k\}$ . This way, a column vector  $\mathbf{x} = [x_i]_n$  is obtained

$$\mathbf{x} = \mathbf{M} \Delta \mathbf{x}^\omega, \quad x_i = \bigwedge_{j=1}^n (m_{ij} + x_j^\omega) \quad (3)$$

### 2.1.2 Morphological Autoassociative Memories *min*

The memories  $\mathbf{W}$  are those that use the maximum product and the minimum operator in their learning phase and the maximum product in their recovery phase.

#### *Learning phase:*

1. For each  $k$ -th element of the fundamental set of associations  $(\mathbf{x}^\mu, \mathbf{x}^\mu)$ , the matrices  $\mathbf{x}^\mu \nabla (-\mathbf{x}^\mu)^\vee$  are calculated.

2. The memory  $\mathbf{W}$  is obtained applying the maximum operator  $\wedge$  to the matrices resulted from step 1.  $\mathbf{W}$  is given by

$$\mathbf{W} = \bigwedge_{\mu=1}^k [\mathbf{x}^{\mu} \nabla (-\mathbf{x}^{\mu})] = [w_{ij}]_{m \times n} \quad (4)$$

$$w_{ij} = \bigwedge_{\mu=1}^k (x_i^{\mu} - x_j^{\mu})$$

**Recovery phase:**

1. The minimum product  $\mathbf{W} \nabla \mathbf{x}^{\omega}$  is calculated, where  $\omega \in \{1, 2, \dots, k\}$ . Then, a column vector  $\mathbf{x} = [x_i]_n$  is obtained

$$\mathbf{x} = \mathbf{W} \nabla \mathbf{x}^{\omega}, \quad x_i = \bigvee_{j=1}^n (w_{ij} + x_j^{\omega}) \quad (5)$$

Theorem 1 in [13] governs the conditions that must be satisfied by both *max* and *min* MAAM to obtain a perfect recall.

**Theorem 1** [13]:  $\mathbf{M} \Delta \mathbf{x}^{\mu} = \mathbf{y}^{\mu}$ ,  $\mathbf{W} \nabla \mathbf{x}^{\mu} = \mathbf{y}^{\mu} \quad \forall \omega = 1, \dots, k$  if and only if for each row index  $i = 1, \dots, m$  there are column indexes  $j_i^{\omega} \in \{1, \dots, n\}$  such that  $m_{ij_i^{\omega}} = y_i^{\omega} - x_{j_i^{\omega}}^{\omega} \quad \forall \omega = 1, \dots, k$ .

### 3 Codebook Generation Based on Morphological Autoassociative Memories

In VQ, vectors of  $k$ -dimension are grouped in a finite set of vectors  $Y = \{y_i : i = 1, 2, \dots, N\}$ . Each vector  $y_i$  is named “codeword”, and the set of all codewords is named “codebook”. The associated neighboring region nearer each codeword is called “Voronoi” region,  $V_i$ . This region is defined as  $V_i = \{x \in R^k : \|x - y_i\| \leq \|x - y_j\|, \forall j \neq i\}$ , where  $x$  represents the input vector.

The codebook design that generates the best representation of an input vector is a very complex process, because it requires an exhaustive search to find the best codebook. Additionally, the search process increases its complexity when the number of codified vectors increases.

The proposed algorithm generates codebooks using MAM. As it was before mentioned, MAM base their operation on the morphological operations, dilation and erosion. Moreover, MAM have low memory requirements; they work with a limited arithmetical precision and a reduced number of simple operations. These MAM features allow to generate a codebook with high search efficiency, high processing speed and robustness to the noise induced in the codewords.

#### 3.1 Codebook Generation Algorithm

Let  $\{I^{\gamma} | \gamma = 1, 2, \dots, h\}$  be  $h$  images defined as the image set used in the codebook generation. This image set is denoted by “ $IS$ ”. Codebook generation process is initialized with an image  $I$  of the set  $IS$ .



Let  $I$  be a matrix,  $\mathbf{E} = [e_{ij}]_{m \times n}$ , where  $m$  represents the height of the image and  $n$  the width of the image. The matrix  $\mathbf{E}$  is divided into sub-matrices of size  $m' \times n'$ , where  $m' = n'$ , obtaining  $N = (m/n') \cdot (n/n')$  sub-matrices; each of these sub-matrices represent a codeword,  $\mathbf{C} = [c_{ij}]_{n' \times n'}$ .

$N$  codewords symbolize the MAAM input vectors. Hence, it is necessary to convert these matrices into vectors:

$$\mathbf{x}^\mu = \mathbf{C}^\mu \mid \mu = 1, 2, \dots, N$$

$$[x_i]_{n'} = [c_{ij}]_{n' \times n'} \tag{6}$$

Once the relation between codewords and input vectors is defined, the codebook generation algorithm is applied. The codebook generation is implemented by applying (2) to every codewords when MAAM *max* are used, or by applying (4) to every codewords when MAAM *min* are used, as it is shown in Fig. 1. In Fig. 1,  $\mathbf{M}$  represent codebook when MAAM *max* is used and  $\mathbf{W}$  represent codebook when MAAM *min* is used.

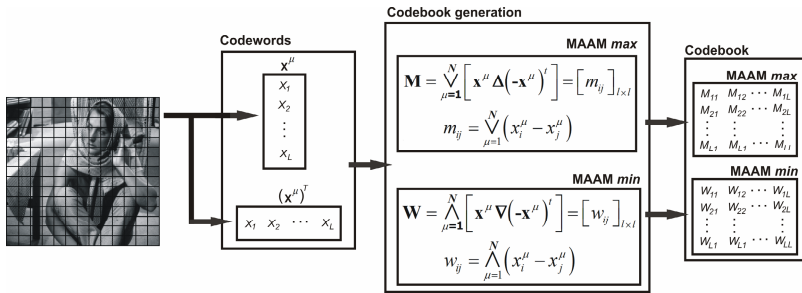


Fig. 1. Codebook generation algorithm

An important parameter in codebook generation is the size of sub-matrices,  $n'$ . The size  $n'$  determines the codewords size and directly affects the codebook dimensions. The choice of  $n'$  depends of image size used in codebook generation. It is recommendable to choose  $n' = 2^p \mid p = 2, 3, 4, 5$  for images of 512x512 pixels. Table 1 shows the parameters obtained with different  $n'$  values. The dimensions of codewords and codebook are defined by:

$$\text{Codeword dimension} = l = n' \cdot n'$$

$$\text{Codebook dimension} = N = (m \cdot n) / l \text{ codewords} \tag{7}$$

The size of sub-matrices  $n'$  determines two important codebook features: image quality and processing speed. The choice of  $n'$  close to 2 generates a codebook that operated at a high processing speed, but the codeword of small dimension leads to the lost of details in the recovered image. On the other hand, choosing  $n'$  close to  $n$  results in a low processing speed in search process, see Table 2. The processing speed for both codebook generation and search process depends on image dimensions and  $n'$  value.

$$\text{Number of operations} = N \cdot l \cdot l = [(m \cdot n) / l] \cdot [l \cdot l] = m \cdot n \cdot l = m \cdot n \cdot n' \cdot n' \tag{8}$$

**Table 1.** Dimension of codewords and codebook for different  $n'$  values

$P$	$n' = 2^P$	Codeword dimension	Codebook dimension
1	2	4	65536
2	4	16	16384
3	8	64	4096
4	16	256	1024
5	32	1024	256
6	64	4096	64
7	128	16384	16

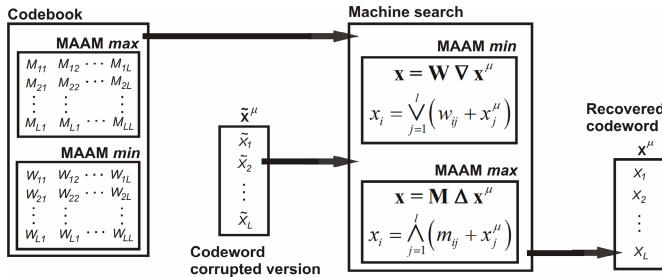
**Table 2.** Processing speed for both codebook generation and search process for images of 512x512 pixels

$P$	$n' = 2^P$	Number of operations for both codebook generation and search process
2	4	4,194,304 sums; 4,194,304 comparisons
3	8	16,777,216 sums; 16,777,216 comparisons
4	16	67,108,864 sums; 67,108,864 comparisons
5	32	268,435,456 sums; 268,435,456 comparisons

Analyzing Theorem 1, one can observe that main diagonal of both  $\mathbf{M}$  and  $\mathbf{W}$  always has value of 0,  $m_{ii} = x_i - x_i = 0$  and  $w_{ii} = x_i - x_i = 0$ . This feature guarantees that the theorem is always fulfilled, granting to the MAAM the property of the maximum capacity of learning. Based on this property, our algorithm is able to perform an evolutionary process because the codified information from new images is added to the codebook without affecting its information already contained. The resulted evolutionary process allows to recover images with better quality without affecting the speed of processing.

### 3.2 Codewords Search Process

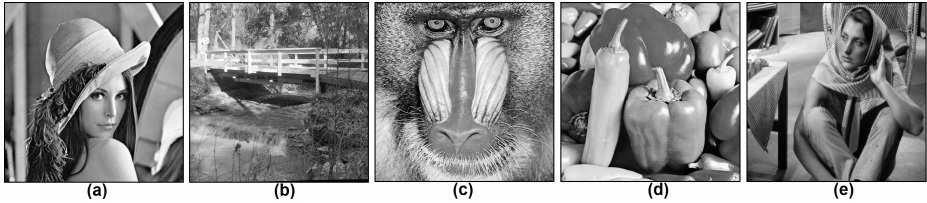
The complement to the codebook generation is the search process to find the codeword codified in the codebook. The search process in our algorithm uses the MAAM recovery phase, as it is shown in Fig. 2. The search process dealing with MAAM *max* uses the minimum product  $\mathbf{M} \Delta \mathbf{x}^\mu$ , and when dealing with MAAM *min*, the search process uses the maximum product  $\mathbf{W} \nabla \mathbf{x}^\mu$ , where  $\mu \in \{1, 2, \dots, N\}$ .



**Fig. 2.** Codeword search process

## 4 Results

In codebook generation, the *IS* set was composed by gray tone images of 512 x 512 pixel resolution shown in Fig. 3.



**Fig. 3.** Images set *IS*. (a) Lena, (b) Bridge, (c) Baboon, (d) Peppers, (e) Barbara.

In order to analyze the operation of the generated codebook, we apply a test process that consists in following steps:

1. Codebook by means of MAAM *min* or MAAM *max* is created.
2. DCT is applied on the image.
3. Vector quantization is applied on DCT coefficients.
4. The inverse processes of quantization and DCT are applied
5. Codewords search process is performed.

In order to have a reference for the results obtained by the proposed algorithm, a test process without the use of the codebook is applied to each image. The results can be observed in Table 3.

Next, we applied the test process in case when only one image was used in codebook generation. Table 4 presents the results of this test, for case when a quantization factor of 32 was used. These results allow to compare the performance of the codebooks generated with each one of set *IS* images.

Table 5 shows the results in case when the codebook was generated with Lena image. Comparing these results to those in Tables 3, it is possible to observe a remarkable improvement in the quality of the restored image that was used for codebook generation, but, the quality of other restored images is poor.

The evolutionary process in codebook appears when the algorithm is applied to other images. This process codifies the information from the new images into the already existed codebook. This evolutionary process improves the codeword search

**Table 3.** Results of applying test process without using codebook

Image	Signal to noise ratio					
	Quantization factor					
	1	2	4	8	16	32
Lena	40.52	35.87	32.62	30.96	29.50	28.04
Bridge	37.34	32.76	30.68	29.07	28.54	26.91
Baboon	35.69	30.62	28.01	26.37	25.51	24.27
Peppers	41.73	36.59	34.69	32.38	30.62	28.86
Barbara	39.99	34.75	32.41	31.15	29.82	28.30

process and generates a robust codebook allowing obtain images of better quality. Tables 6 and 7 show the codebook performance when the codebook was generated using several images. These results show that with the evolutionary process, the image quality increases when a new image is codified into the codebook.

The performance plots of the codebook based on both MAAM *min* and MAAM *max* and applied to Lena and Barbara images are shown in Fig. 4 and Fig. 5.

**Table 4.** Results of test process for codebooks generated with each image of set *IS*

Image	Signal to noise ratio (dB, quantization factor = 32)											
	Image used in codebook generation											
	Lena		Bridge		Baboon		Peppers		Barbara			
	MAAM		MAAM		MAAM		MAAM		MAAM			
	<i>min</i>	<i>max</i>	<i>Min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>		
Lena	29.47	28.38	26.23	25.44	28.12	27.52	28.29	27.31	27.82	26.98		
Bridge	26.69	27.21	26.93	27.52	26.88	27.44	26.74	27.34	26.72	27.29		
Baboon	23.21	22.29	23.25	22.23	25.11	24.53	23.60	22.64	26.60	22.61		
Peppers	28.93	27.41	27.98	25.49	29.77	28.23	30.49	28.91	29.40	27.56		
Barbara	27.05	27.40	25.91	26.37	28.24	28.37	26.96	27.10	28.77	28.66		

**Table 5.** Results of test process when codebook is generated with Lena image

Image	Signal to noise ratio (dB)											
	Quantization factor											
	1		2		4		8		16		32	
	MAAM		MAAM		MAAM		MAAM		MAAM		MAAM	
	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>
Lena	42.97	41.10	37.76	36.69	34.39	33.15	32.60	31.51	30.99	29.89	29.47	28.38
Bridge	34.02	35.02	31.34	32.19	29.81	30.82	28.49	29.43	28.11	28.88	26.69	27.21
Baboon	24.66	24.20	24.35	23.73	24.02	23.24	23.74	22.83	23.59	22.66	23.21	22.29
Peppers	32.91	31.18	32.07	30.44	31.55	29.95	30.79	29.11	29.96	28.32	28.93	27.41
Barbara	30.71	31.45	29.73	30.52	28.95	29.94	28.48	29.28	27.85	28.47	27.05	27.40

**Table 6.** Results of test process when codebook is generated using Lena, Baboon and Peppers images

Image	Signal to noise ratio (dB)											
	Quantization factor											
	1		2		4		8		16		32	
	MAAM		MAAM		MAAM		MAAM		MAAM		MAAM	
	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>
Lena	42.08	41.11	36.87	36.40	33.56	33.02	31.87	31.38	30.46	29.80	28.91	28.33
Bridge	37.29	37.50	32.74	32.87	30.67	30.76	29.06	29.26	28.53	28.65	26.90	26.98
Baboon	37.31	36.21	31.78	31.11	29.16	28.40	27.49	26.71	26.19	25.82	24.63	24.43
Peppers	42.26	41.73	36.99	36.67	35.37	34.69	33.05	32.38	31.48	30.62	29.53	28.94
Barbara	37.30	38.38	33.82	34.63	31.88	32.73	30.87	31.46	29.59	30.00	28.22	28.41

The before presented results were obtained using the size of sub-matrices  $n'=16$  in codebook generation. Being based on the realized experiments, we consider that this size  $n'=16$  offers a balance point between processing speed and image quality. Table 2 shows as the size of sub-matrices  $n'$  affects both the processing speed of codebook generation and processing speed of the codeword search process. The value of  $n'$  also affects the recovered image quality, Table 8 shows as the PSNR is affected when  $n' =$

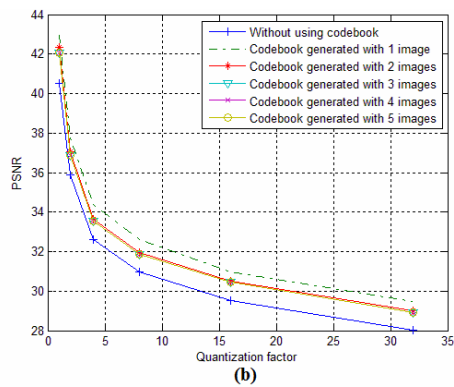
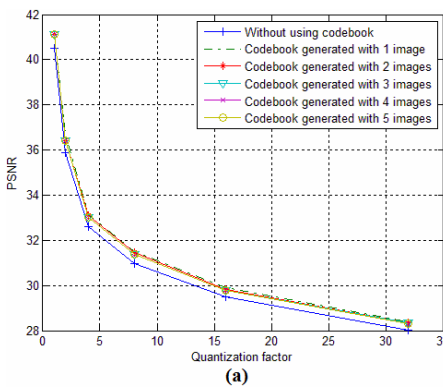
4, 32. The codebook performance based on both MAAM *min* and MAAM *max* and varying the value of  $n'$  applied on Lena and Barbara images are plotted in Figures 6 and 7.

**Table 7.** Results of test process when codebook is generated with all images of set *IS*

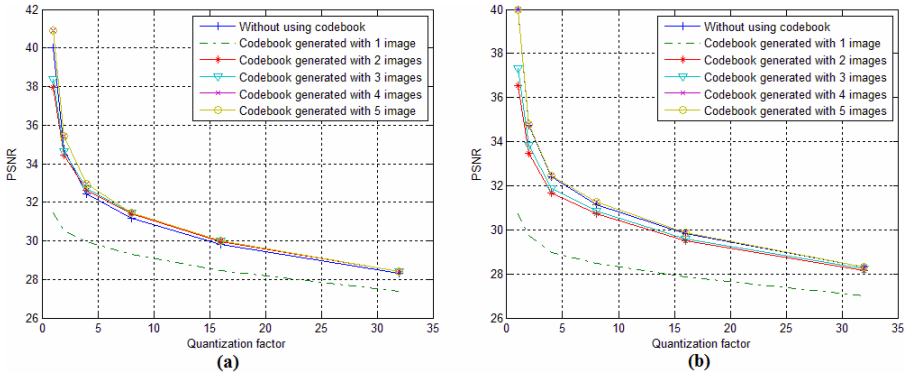
Image	Signal to noise ratio (dB)											
	Quantization factor											
	1		2		4		8		16		32	
	MAAM <i>min</i>	MAAM <i>max</i>	MAAM <i>min</i>	MAAM <i>max</i>	MAAM <i>min</i>	MAAM <i>max</i>	MAAM <i>min</i>	MAAM <i>max</i>	MAAM <i>min</i>	MAAM <i>max</i>	MAAM <i>min</i>	MAAM <i>max</i>
Lena	42.08	41.11	36.87	36.40	33.53	33.03	31.87	31.38	30.46	29.80	28.91	28.31
Bridge	37.34	37.43	32.76	32.78	30.68	30.73	29.07	29.11	28.54	28.55	26.91	26.91
Baboon	37.14	36.21	31.70	31.02	29.07	28.37	27.34	26.66	26.04	25.81	24.58	24.42
Peppers	42.26	41.73	36.99	36.67	35.38	34.69	33.05	32.38	31.43	30.62	29.48	28.94
Barbara	39.99	40.91	34.81	35.44	32.44	32.94	31.26	31.47	29.85	30.01	28.32	28.43

**Table 8.** Results of test process when codebook is generated with  $n' = 4, 32$

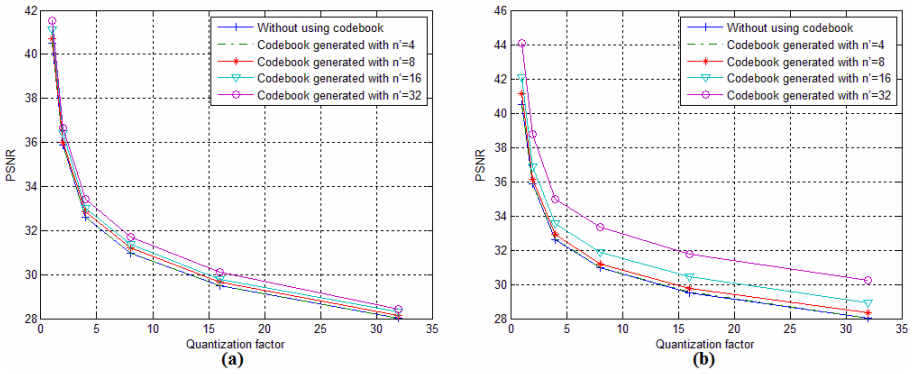
$n'$	Image	Signal to noise ratio (dB)											
		Quantization factor											
		1		2		4		8		16		32	
		MAAM <i>min</i>	MAAM <i>max</i>	MAAM <i>min</i>	MAAM <i>max</i>	MAAM <i>min</i>	MAAM <i>max</i>	MAAM <i>min</i>	MAAM <i>max</i>	MAAM <i>min</i>	MAAM <i>max</i>	MAAM <i>min</i>	MAAM <i>max</i>
4	Lena	40.52	40.52	35.87	35.87	32.65	32.62	30.96	30.98	29.53	29.50	28.05	28.04
	Bridge	37.34	37.34	32.76	32.75	30.68	30.67	29.07	29.05	28.54	28.51	26.91	26.88
	Baboon	35.82	35.75	30.68	30.62	28.07	28.04	26.39	26.38	25.52	25.51	24.28	24.27
	Peppers	41.73	41.73	36.59	36.59	34.69	34.69	32.38	32.38	30.64	30.62	28.86	28.86
	Barbara	39.99	39.99	34.75	34.75	32.41	32.39	31.15	31.14	29.82	29.80	28.30	28.28
32	Lena	44.10	41.51	38.75	36.65	34.95	33.42	33.33	31.69	31.75	30.09	30.24	28.43
	Bridge	37.34	38.34	32.76	33.58	30.68	31.83	29.07	30.09	28.53	29.34	26.90	27.49
	Baboon	39.78	36.63	34.54	31.45	31.29	28.78	29.23	27.04	27.77	26.04	25.57	24.61
	Peppers	43.93	42.46	39.31	37.01	37.22	34.98	34.89	32.56	32.98	30.75	31.27	28.95
	Barbara	40.53	43.61	35.22	36.93	32.70	34.95	31.50	33.01	30.16	30.99	28.49	29.02



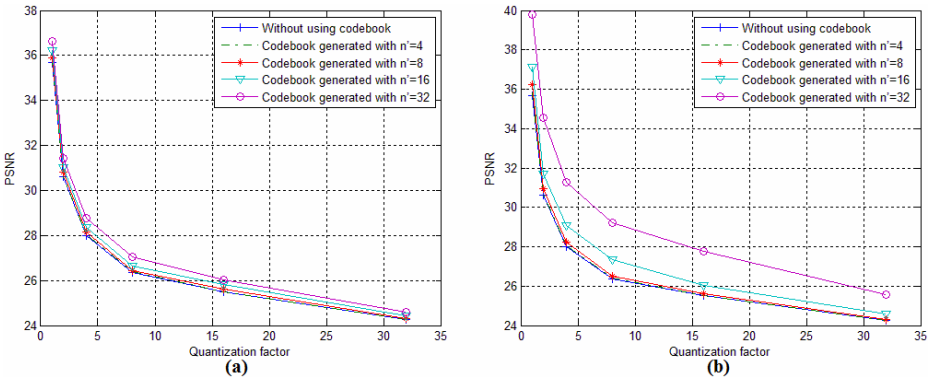
**Fig. 4.** Performance of codebooks based on Lena image, (a) codebook generated with MAAM *max*, (b) codebook generated with MAAM *min*



**Fig. 5.** Performance of codebooks based on Barbara image, (a) codebook generated with MAAM max, (b) codebook generated with MAAM min



**Fig. 6.** Comparison of codebooks performance on Lena image, (a) codebook generated with MAAM max, (b) codebook generated with MAAM min



**Fig. 7.** Comparison of codebooks performance on Baboon image, (a) codebook generated with MAAM max, (b) codebook generated with MAAM min

## 5 Conclusions

The use of MAAM in both codebook generation and codeword search process has demonstrated a high efficiency in the processing speed and recovered image quality. The proposed codebook generation algorithm allows adding information from new images, obtaining an evolution in the codeword search process that results in better quality of the recovered image without affecting the processing speed. The size of sub-matrices  $n'=16$  in codebook generation offers a balance point between processing speed and image quality.

## Acknowledgements

This work was supported by Instituto Politécnico Nacional as a part of the research project SIP#20071380.

## References

1. Linde, Y., Buzo, A., Gray, R.: An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications* 28(1), 84–95 (1980)
2. Huang, C.M., Harris, R.W.: A Comparison of Several Vector Quantization Codebook Generations Approaches. *IEEE Trans. on Image Proce.* 2(1), 108–112 (1993)
3. Equitz, W.H.: A New Vector Quantization Clustering Algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing* 37(10), 1568–1575 (1989)
4. Vaisey, J., Gersho, A.: Simulated Annealing and Codebook Design. In: *IEEE Proc. ICASSP 1988*, pp. 1176–1179 (1988)
5. Flanagan, J.K., Morrell, D.R., Frost, R.L., Read, C.J., Nelson, B.E.: Vector Quantization Codebook Generation Using Simulated Annealing. *IEEE Proc.*, 1759–1762 (1989)
6. Kohonen, T.: Automatic formation of topological maps of patterns in a self-organizing system. In: Oja, E., Simula, O. (eds.) *Proc. 2SCIA, Scand. Conf. on Image Analysis*, Helsinki, Finland, pp. 214–220 (1981)
7. Amerijckx, C., Verleysen, M., Thissen, P., Legat, J.-D.: Image Compression by Self-Organized Kohonen Map. *IEEE Trans. on Neural Networks* 9(3), 503–507 (1998)
8. Amerijckx, C., Legat, J.-D., Verleysen, M.: Image Compression Using Self-Organizing Maps. *Systems Analysis Modelling Simulation* 43(11), 1529–1543 (2003)
9. Ritter, G.X., Sussner, P., Díaz de León, J.L.: Morphological Associative Memories. *IEEE Trans. on Neural Networks* 9(2), 281–293 (1998)
10. Yáñez y, C., Díaz de León, J.L.: *Memorias Morfológicas Heteroasociativas*. CIC, IPN, México, IT 57, Serie Verde (2001), ISBN 970-18-6697-5
11. Yáñez y, C., Díaz de León, J.L.: *Memorias Morfológicas Autoasociativas*. CIC, IPN, México, IT 58, Serie Verde (2001), ISBN 970-18-6698-3
12. Guzman, E., Pogrebnyak, O., Yáñez, C., Moreno, J.A.: Image Compression Algorithm Based on Morphological Associative Memories. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) *CIARP 2006*. LNCS, vol. 4225, pp. 519–528. Springer, Heidelberg (2006)
13. Ritter, G.X., Díaz de León, J.L., Sussner, P.: Morphological Bidirectional Associative Memories. *Neural Networks* 12(6), 851–867 (1999)

# A Shape-Based Model for Visual Information Retrieval

Alberto Chávez-Aragón<sup>1</sup> and Oleg Starostenko<sup>2</sup>

<sup>1</sup> Universidad Autónoma de Tlaxcala, Calzada Apizaquito s/n km. 1.5,  
Apizaco Tlaxcala, México

<sup>2</sup> Universidad de las Américas-Puebla, Sta. Catarina Mártir Cholula, Puebla, México

**Abstract.** This paper presents a novel shape-based image retrieval model. We focused on the shape feature of objects inside images because there is evidence that natural objects are primarily recognized by their shapes. Using this feature of objects the semantic gap is reduced considerably. Our technique contains an alternative representation of shapes that we have called two segment turning function (2STF). Two segment turning function has a set of invariant features such as invariant to rotation, scaling and translation. Then, based on 2STF, we proposed a complete new strategy to compute a similarity among shapes. This new method was called Star Field (SF). To test the proposed technique, which is made up of a set of new methods mentioned above, a test-bed CBIR system was implemented. The name of this CBIR System is IRONS. IRONS stands for "Image Retrieval based ON Shape". Finally, we compared our results with a set of well known methods obtained similar results without the exhaustive search of many of them. This former feature of our proposal is one of the most important contributions of our technique to the visual information retrieval area.

## 1 Introduction

Most of the current search engines' image retrieval algorithms use text as a principal document descriptor. Techniques which use different descriptors like shape, color, sound, etc. lag behind text-based techniques. Thus, there is a growing need for efficient visual information retrieval algorithms which go beyond the text-based retrieval approach. This paper addresses the problem of retrieving documents that contain visual information. We specifically proposed a new technique for the image retrieval problem based on shape, since shape has a meaning by itself. On the other hand, an extension of the ontology concept, which is used in the information retrieval based on text area, is proposed in the image domain. Using ontologies in an image-restricted domain makes possible the reduction of nonsense results.

### 1.1 Shape-Based Retrieval

Perhaps the most obvious requirement of users for VIR systems is to retrieve images by shape, since there is evidence that natural objects are primarily recognized by their shape [5]. Features vectors which represent object shapes contained



in images are computed in order to be indexed in a database. The query process works in the same way that color-based and texture-based retrieval work in the sense that a query can be an image. But, unlike color and texture retrieval, shape-based retrieval has another particular way to feed the query into the system. This is by means of sketching. Systems which support this kind of queries must provide the user with a sketch tool [6], [14].

## 2 Shape Representation

Traditionally, a shape is described as a closed polygon. However, the polygonal representation of shapes is not a convenient way to compute the similarity among them. In order to overcome this problem we propose a different representation that we have called two-segment turning function. Our technique is based on tangent space representation but it has some advantages that are outlined below.

Our strategy to compute the similarity among shapes starts out getting the outline of the shape from an image. Basically, we assume as a premise that each image we are working with represents just one object. Besides, the object has been previously separated from the background. That means that our images are binary ones and the objects are represented by white pixels and the backgrounds by black pixels. Using a simple outline detector algorithm and transforming the object outline into a closed polygon we can simplified the problem focusing on a polygonal matching task. However, these polygons have plenty of vertices. The next natural step is to reduce the number of vertices so that we can apply an efficient similarity strategy. The process which allows us to reduce the number of vertices maintained the important ones is called evolution of polygons or curve evolution.

### 2.1 Relevance Measure for Polygonal Vertices

In order to decrease the number of vertices of a shape it is necessary to calculate what is the relevance of each vertex. The relevance measure  $K$  that we use is based on two parameters, the length and the turn angle of two consecutive line segments which share the vertex we want to compute its relevance. The relevance is defined as it is shown in equation [1].

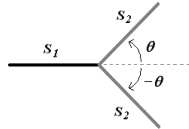
$$K(S_1, S_2) = \frac{\beta(S_1, S_2)l(S_1)l(S_2)}{l(S_1) + l(S_2)} \quad (1)$$

where  $\beta(S_1, S_2)l(S_1)$  is the turn angle at the common vertex of the segments  $S_1, S_2$ , and  $l$  is the length function normalized with respect to the total length of the polygonal curve  $C$ . The lower value of  $K(S_1, S_2)$  is, the less contribution to the shape of the curve of arc  $S_1 \cup S_2$  is. To stop the evolution process it is necessary to use a parameter that defines the number of iterations or to use

a threshold which represents the permitted range of values for any simplified shape vertex.

### 2.2 Two-Segment Turning Function

Using two-segment turning function or *2STF* a polygonal curve  $P$  is represented by the graph of a step function, the steps on  $x - axis$  represents the normalized arc length of each segment in  $P$ , and the  $y - axis$  represents the turn angle between two consecutive segments in  $P$ . The former feature gives the name to our proposed technique. Figure 1 shows the angle that is taking into account in order to build the *2STF*. The angle  $\theta$  is defined by  $S_2$  and the imaginary line that pass through the segment  $S_1$ . This way of measuring the angle has an intuitive reason and this is that the angle  $\theta$  measures the deviation of the second segment with respect to the first segment direction. It is clear that the angle values are in the interval  $[-\pi, \pi]$ .



**Fig. 1.** This figure shows the angle used for *2STF*. The angle is defined by the imaginary line passing through the first segment and the second one. A left turn makes the angle positive and a turn in the clockwise direction makes the angle negative.

## 3 Star Field Representation

Star Field (*SF*) is our alternative representation for shapes that allows us to apply a different algorithm in order to obtain a similarity value of two curves. This new algorithm that we will propose below does not provide a way to determine the best correspondence among two functions but a very good solution. As a result, Star Field along with a new similarity algorithm are expected to give an easier and faster matching process. A Star Field formally is a torus  $T_1 \times T_2$ , where  $T_1$  is a circle of length one that represents the length of a polygonal curve and  $T_2$  is a circle that represents the turning direction of digital steps from *2STF*. Nevertheless, most of the time we consider a *SF* as a window that shows a *2D* projection of the torus. This window is made up of stars or points, that is where the name comes from, and each of them represents the relevance measure of each *2STF* step.

### 3.1 From *2STF* to *SF*

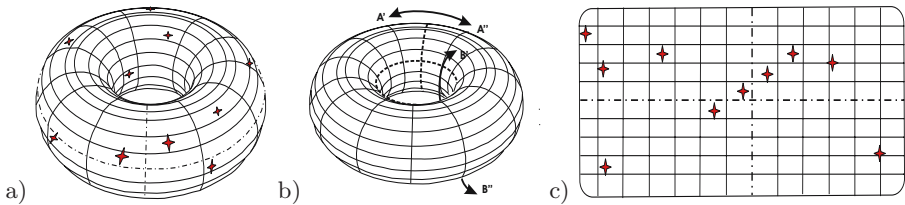
One of the mayor difference between the use of *2STF*'s similarity measure an the one using *SF* is regarding to the grade of evolution of the digital curves they work

with. A star field diagram is basically a  $2D$  plane, it is divided horizontally into two sections. The upper section holds the stars that represent vertices of concave arcs. On the other hand, lower part holds vertices of convex arcs. Each star on  $SF$  is defined by means of two coordinates. The  $y$  – coordinate represents the angle between two consecutive segments. Due to the use of  $2STF$  to represent a shape, the interval of the turning angle is  $[-\pi, \pi]$  radians. However, in the Star Field the angle is normalized in the interval  $[0, 1]$ . With respect to the  $x$  – coordinate, these values correspond to the accumulative length of the steps in  $2STF$  from the starting point to the current point. In other words, the  $x$  – coordinates represent how far is each vertex from the starting vertex and also this distance is normalized.

To illustrate the way a Star Field looks like, imagine that the  $2STF$  has just decreasing steps, the Star Field representation of this function will be crowded in the lower part. This kind of Star Fields represents mainly convex shapes. In the same way, if the  $2STF$  shows raising steps, that means that it represents a mainly concave figure and the Star Field is crowded in the upper part. Finally, if a step has an angle equal to zero with respect to the previous one, the  $y$  – coordinate of the corresponding star has the value .5 in the Star Field, this is because the values of the Star Field go from  $[0,1]$  in both directions. Likewise, if two consecutive segments have  $-\pi$  or  $\pi$  radians the  $y$  – coordinate of the corresponding point in the Star Field has 0 or 1 respectively. To illustrate this, consider figure 2.

As we have mentioned before, a Star Field diagram is basically a  $2D$  plane. In order to transform a torus into a  $2D$  plane, we imaginatively cut the torus on two places following dotted-lines as it is shown in figure 2 b). Then, it is necessary to bend the surface, in the sense the arrows show, to get our desired  $2D$  plane. As a result, we obtained a plane similar to the one shown in figure 2 c).

Since Star Field is based on  $2STF$ , it has the same invariant characteristics as  $2STF$ , demonstration of those features are beyond the scope of this paper, for further details see [2].



**Fig. 2.** a). Star Field, real representation. Actually, Star Fields can be seen as a bending surface like it is shown in this figure. Each star or point in the Star Field represents the vertex that is shared by two consecutive steps from the equivalent  $2STF$ . b). This figure shows where to cut the torus and in what direction we have to bend it, so that a  $2D$  Star Field representation is obtained. c).  $2D$  Star Field representation.

## 4 Matching Graph

We proposed a new similarity measure that makes use of a graph that has particular features. In this section the construction process of this graph is presented.

Given two polygonal curves  $P_1$  and  $P_2$  and their Star Field representations  $SF_1$  and  $SF_2$ , the graph  $G$  that allows us to compute their similarity is defined as follows.  $G = (V, E)$  where  $V$  and  $E$  are disjoint finite sets. We call  $V$  the vertex set and  $E$  the edge set of  $G$ . Our particular graph  $G$  has a set  $V$  which consists of two smaller subset of vertices  $v_1$  and  $v_2$ .  $V = v_1 \cup v_2$ , where  $v_1$  is the set of point of  $SF_1$  and  $v_2$  is the set of points of  $SF_2$ . On the other hand,  $E$  is the set of pairs  $(r, s)$ , where  $r \in v_1$  and  $s \in v_2$ .

According to previous definition the edges of our graph, that we will call from now on matching graph or  $MG$ , consists of two points and each point comes from a different Star Field representation. But also a new restriction will must be introduced, this is stated as follows.  $\forall(r, s) \in E$ , there is not more that one pair  $(r, s)$  that has the same point  $s$ . This restriction has an intuitive idea and this is, one point of the first curve can be matched with  $n$  points of the second one but not in the inverse sense. We have to say that the number of points of each Star Field can be different and that is because we can match polygons with different grade of evolution.

### 4.1 Matching Graph Construction

The main idea behind the construction of the matching graph consists in building a connected weighted graph so that an algorithm to find the minimal spanning tree is applied. The minimum spanning tree is a subset of edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. This way, the lower value of total weight the more similar are the shapes involved. But, in order to get the desired result the matching graph must be constructed in a very particular way. This method of construction is shown in the Matching graph construction algorithm .

Given two identical shapes with the same number of steps, the total weight of the spanning tree is equal to zero. This is, because each star is connected with the corresponding one and since they have the same value of  $x - coordinate$  and  $y - coordinate$  the euclidian distance is equal to zero. Additionally, we have mentioned that all the stars from the first shape are connected with a weight equal to zero. As a result, the values of the path through the spanning tree is zero, that means that they are identical. To find the minimum spanning tree, we used the most popular algorithm for this task, the Prim's algorithm.

### 4.2 Similarity Measure

Finally, we can define how to calculate the similarity among shapes. The most important part of this calculation is the value of the cumulative weight of the edges that make up the spanning tree. However, the similarity value is also affected by a penalty quantity, this is because some stars have not been connected

Matching graph construction

**input:** two set of points  $SF_1$  and  $SF_2$  that define the two Star Field representations, an increment  $\Delta$  and a distance  $d$

**output:** a connected weighted graph

1. rotate in the  $x$  direction  $SF_1$  and  $SF_2$  so that, the most import star of each  $SF$  coincides in the center of the window
2. for each point  $sf_1pn$  from the  $SF_1$  do
3. look for those points that belong to  $SF_2$ , that stay at most a distance  $d$  in all directions from  $sf_1pn$  and that have not been connected previously
4. connect  $sf_1pn$  with each point found in previous step and assign a weigh equal to the euclidian distance of the two vertices to each edge
5. if there wasn't any connection, increase  $d$  in a value  $\Delta$  and go to step 3
6. Select one point of  $SF_1$  and connect the rest of the points from  $SF_1$  with it; finally assign each edge generated in this step a weigh equal to zero

with the corresponding ones. So, the penalty quantity is the sum of the distances between each none-connected star and the medial axis of the star field diagram.

## 5 Results

In the majority of the experiments of this paper we used the database CE-Shape-1 [12], [15]. The reason why we selected this image database is because this set of images has been used for testing similar works, this allows us to have a reference framework to compare with. The Core Experiment CE-Shape-1 for shape descriptors performed for the MPEG-7 standard consists of 1400 images divided into 70 classes with 20 images each. A single image is a simple pre-segmented shape defined by their outer closed contour.

Table 1 describes shortly a set of shape descriptors which were tested in Core Experiment CE-Shape-1 and these works are the ones we compare with our proposed method.

First experiment consists in verifying how robust is our method with respect to scaling and rotation changes. We have done this experiment in the way described in part A of MPEG-7 standard experiments. Results are shown in table 2. Our method is labeled as **G**.

We can say that our method is robust to changes in scaling and rotation as we have already demonstrated comparing our method with those of the MPEG-7 core experiment. We cannot forget that the 91.40% was obtained in a very strict experiment and this value is not far from those reported by the the MPEG-7 core experiment and in some cases even better.

**Table 1.** Shape descriptors which were tested in the Core Experiment CE-Shape-1

descriptor	Technology
A	based on the curvature scale-space [13], [4]
B	based on wavelet representation of object contours [3]
C	best possible correspondence of visual parts [9], [10]
D	based on Zernike moments [8]
E	based on multilayer eigenvectors [12]
F	tree-matching algorithm [7], [11], [1]
G	<b>Star fields</b>

**Table 2.** Robustness to scaling and rotation, our results are labeled as G

Shape descriptor	Robustness to scaling and rotation
A	94.56
B	92.75
C	94.32
D	96.07
E	96.21
F	85
G	<b>91.40</b>

## 5.1 IRONS

IRONS is the name of our proposed image retrieval system. The latter stands for: **I**mage **R**etrieval based **ON** **S**hape, since shape is the most important feature we took into account to propose our technique. The image retrieval system consists of three main components (1) an image analysis system which indexes items according to their visual features, (2) a query engine and (3) interface layer that deals with browsing and query. IRONS was developed using the Matlab@language, version 6.5.0.

## 5.2 Retrieval Effectiveness

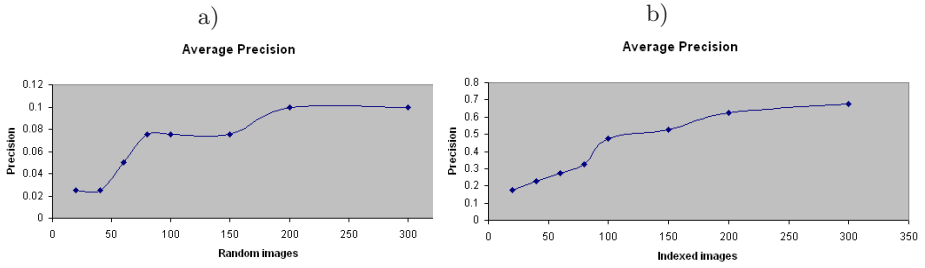
The evaluation of an information retrieval system is non-trivial task. This is because there is an amount of subjectivity involved in deciding the correct result set for a query. The similarity between a query and a set of image results depends on individual perception of user. Nevertheless, there is a standard way of judging the results of a query. This process is through the calculation of two measures associated with information retrieval system: Precision and Recall.

**Candidate Images Selected Randomly** We divided the experiments into three groups. In the first experiment we applied our Star Field Technique over a subset of images from the CE-Shape-1 database. These images were selected randomly. The idea of this experiment is to observe how well the Star Field technique is able to choose the images that belong to the same class of the query. **Candidate Images Selected by means of Structural Features** Second experiment consists in applying our shape retrieval approach over a subset of images that were previously selected using Eccentricity and Solidity. Using structural features the search tree is pruned but not randomly as we presented in the former experiment. Another difference with respect to the first experiment is the fact that there is not guarantee that every image belonging to the same class of the query is present in the candidate image set. **Candidate Images Selected using an Image Ontology** The third experiment consists in pruning the search tree using a technique that is traditionally used in the textual information retrieval area. So we propose to use an ontology in the image domain and as a result we can obtain the advantages from the two areas. An efficient approach which use a shape as a principal descriptor and on the other hand, an ontology that allow us to reduce the semantic gap problem produce a complete new approach.

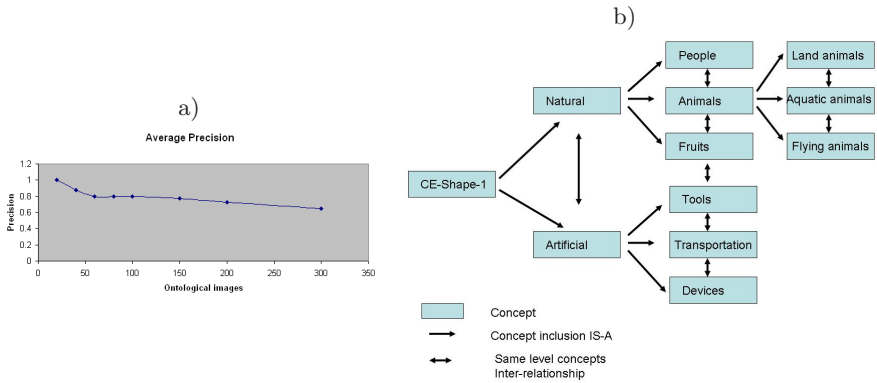
Figure 3 a) shows the average precision for the first experiment. X-axis shows the number of images used as a preliminary set of search, Y-Axis shows the precision. When our method is applied over a reduced set of images, precision is low. When the number of images increases the precision also increases because there are more possibilities of applying the similarity metric over a major number of shapes that belong to the image query's class. The average precision for this experiments is 0.065. With respect to the recall, it shows a similar behavior that precision figure, this is because, the number of retrieved and displayed images is equal to the number of relevant images for each query, that is 20 images.

Results of the second experiment are shown in figure 3 b). The time required to run the algorithms is similar that the one presented in previous experiment due to the amount of images per experiment is the same that the previous one. The average precision and recall of this experiment is 0.412.

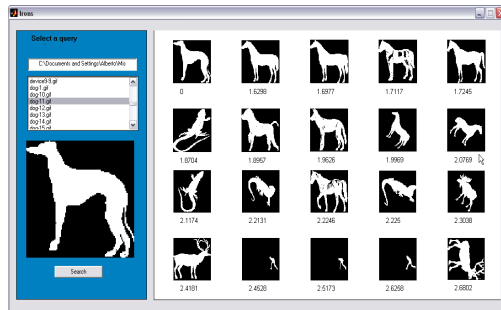
In the third experiment the extraction of the subset of candidate images to be applied Star field metrics is the following. When the user queries a land animal, lets say a dog, for instance; the set of candidate images is made up of images from class dog and those images that contains horses, bears, deers, etc. Even thought, those images have not been labeled as dogs. But they are at the same level and the user finds a major grade of satisfaction because the semantic gap



**Fig. 3.** a). Average precision when the Star Field technique is applied over a random set of image. b). Average Precision when the Star Field technique is applied over a group of candidate images that share structural features.



**Fig. 4.** a). Average Precision of experiment which uses ontological images b). Proposed Ontology for the Core Experiment Shape - 1.



**Fig. 5.** Example of results from experiment one



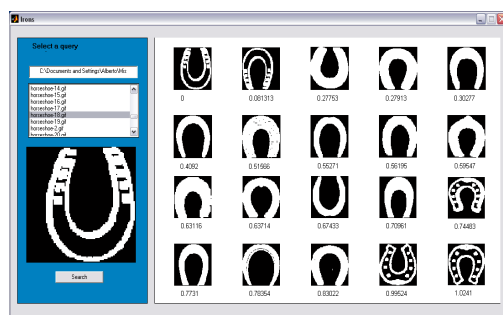


Fig. 6. Example of results from experiment three

has been reduced. Another advantage is that the total images belonging to the image query are presented in the candidate image set. As a consequence, the results are better as figure 4 a) shows. In this experiment the average precision value is 0.803. We organized the set of images from Core Experiment-Shape-1 in semantic categories. Each class is classified by means of the ontology shown in figure 4 b).

Figures 5, 6 show examples of experiments one, and three respectively.

## 6 Conclusions

We proposed a complete new strategy to compute the similarity among shapes. This new technique was called Star Field (*SF*). Star Field inherits from *2STF* invariant characteristics. Additionally, Star Field allows us to work with less simplified digital polygons; since, it permits to define a similarity measure based on the calculation of a minimum spanning tree from a connected weighted graph. Among the outstanding points of our technique we can mention: ease to use and implement, it uses visual parts as a parameter of similarity like humans do, it has a good performance as we demonstrated in this paper. The proposed technique, which is made up of a set of new methods, was implemented in a test-bed CBIR system that we called "Image Retrieval based ON Shape". To conclude, we proposed a high effective, ease to implement and robust image retrieval technique which uses the shapes of the objects as a main descriptor. Our approach is comparable in results with those systems which compute the best correspondence among shapes. However, our approach does not attend to find the best correspondence but it finds a very good approximation.

## References

1. Blum, H.: Biological shape and visual science. *Journal of Theor. Biol.* 38, 205–287 (1973)
2. Chavez-Aragon, J.A.: Star Field Approach for Shape-Based Image Retrieval: Development, Analysis and Applications. Ph.d. Thesis, Universidad de las Americas-Puebla, Cholula Puebla Mexico (2007)

3. Chuang, G., Kuo, C.-C.: Wavelet descriptor of planar curves: Theory and applications. *IEEE Trans. on Image Processing* 5, 56–70 (1996)
4. Mokhtarian, S.A.F., Kittler, J.: Efficient and robust retrieval by shape content through curvature scale space. In: Smeulders, A.W.M., Jain, R. (eds.) *Image Database and Multimedia Search*, pp. 51–58. World Scientific Publishing, Singapore (1997)
5. Biederman, I.: Recognition-by-components: a theory of human image understanding. *Psychological Review* 94(2), 115–147 (1987)
6. Hirata, K., Kato, T.: Query by visual example - content-based image retrieval. In: *EDBT 1992 Third International Conference on Extending Database Technology*, vol. 1, pp. 56–71 (1992)
7. Siddiqi, S.J.D.K., Shokoufandeh, A., Zucker, S.W.: Shock graphs and shape matching. *Int. J. of Computer Vision* (2000)
8. Khotanzan, A., Hong, Y.H.: Invariant image recognition by zernike moments. *IEEE Trans. PAMI* 12, 489–497 (1990)
9. Latecki, L.J., Lakämper, R.: Contour-based shape similarity. In: Huijismans, D.P., Smeulders, A.W.M. (eds.) *VISUAL 1999. LNCS*, vol. 1614, pp. 441–454. Springer, Heidelberg (1999)
10. Latecki, L.J., Lakämper, R.: Shape similarity measure based on correspondence of visual parts. *IEEE Trans. Pattern Analysis and Machine Intelligence* (2000)
11. Lin, I.-J., Kung, S.Y.: Coding and comparison of dags as a novel neural structure with application to on-line handwritten recognition. *IEEE Trans. Signal Processing* (1996)
12. Longin, R.L., Latecki, J., Eckhardt.: Shape descriptors for non-rigid shape with a single closed contour. *CVPR 2000* (2000)
13. Mokhtarian, F., Mackworth, A.K.: A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. PAMI* 14, 789–805 (1992)
14. Chan, Y., Kung, S.Y.: A hierarchical algorithm for image retrieval by sketch. *First IEEE Workshop on Multimedia Signal Processing* 1, 564–569 (1997)
15. Ye, J., Smeaton, A.F.: Aggregated Feature Retrieval for MPEG-7. In: Sebastiani, F. (ed.) *ECIR 2003*, vol. 2633 (2003)

# An Indexing and Retrieval System of Historic Art Images Based on Fuzzy Shape Similarity

Wafa Maghrebi<sup>1</sup>, Leila Baccour<sup>1</sup>, Mohamed A. Khabou<sup>2</sup>, and Adel M. Alimi<sup>1</sup>

<sup>1</sup> REsearch Group on Intelligent Machines (REGIM) University of Sfax,  
ENIS, DGE, BP. W-3038, Sfax, Tunisia

wafa.maghrebi@fsegs.rnu.tn, leila.baccour@edunet.tn,  
adel.alimi@ieee.org

<sup>2</sup> Electrical and Computer Engineering Dept, University of West Florida,  
11000 University Parkway, Pensacola, FL 32514, USA  
mkhabou@uwf.edu

**Abstract.** We present an indexing and retrieval system of historic art images based on fuzzy shape similarity. The system is composed of three principal components: object annotation, object shape indexing, and query/retrieval. The object annotation in database images is done manually offline. The object shape indexing and retrieval, however, are done automatically. Annotated object shapes are indexed using an extended curvature scale space (CSS) descriptor suitable for concave and convex shapes. The query/retrieval of pertinent shapes from the database starts with a user drawing query (with a computer mouse or a pen) that is compared to entries in the database using a fuzzy similarity measure. The system is tested on a set of complex color and grey scale images of ancient documents, mosaics, and artifacts from the National Library of Tunisia, the National Archives of Tunisia, and a selection of Tunisian museums. The system's recall and precision rates were 83% and 60%, respectively.

**Keywords:** Image indexing, image retrieval, eccentricity, circularity, curvature space descriptors, fuzzy shape similarity.

## 1 Introduction

The National Library of Tunisia, the National Archives of Tunisia [1, 2], and a selection of Tunisian museums (e.g. Bardo, El-Jem, Enfidha, Sousse, Sfax) contain a huge selection of ancient documents, mosaics, and artifacts of important historic value. These treasures are carefully photographed and catalogued to make them available to researchers, but also to limit direct handling of these fragile articles. Color and grey scale images of these articles are available in many databases including the one used to test the system presented in this paper. A system that allows a user to search the images in these databases using either a sample or a user-drawn shape is very desirable and useful [3, 5]. The images are very rich with complex content consisting of many objects of different shape, color, size, and texture which makes the automatic extraction of meaningful objects from an image very challenging

[5] if not impossible (Fig. 1). The system we are proposing does not attempt to automatically extract objects from an image (objects are extracted manually by outlining their contours), however it does provide an automatic, robust, and efficient way to index extracted object shapes and retrieve similar objects from the database.

In section 2 we give an overview of the proposed system architecture and describe in details each of its components. Section 3 describes the results of our system evaluation. The conclusion is presented in section 4.



**Fig. 1.** Sample of mosaic color images

## 2 System Description

Our system is composed of three major modules: object annotation module, object indexation module, and database query and retrieval module. Figure 2 shows an overview of the system and the following subsections describe each module in details.

### 2.1 Annotation Module

The annotation module of our system was inspired by the Vindx system [6, 7] which relies on a user's perception to manually annotate areas of interest and important shapes/objects in a database of 17th century paintings. The images are manually indexed based on the shapes they contain. Our annotation module consists of a graphical user interface that presents a user with a list of images to be annotated (Fig. 3). The user selects an image of his/her choice and, with a computer mouse or stylus, can trace the contours of areas of interests and/or important objects in the image. The user can also associate textual annotations (e.g. person, animal, vegetation, graphics, etc.) with that image (Fig. 4). The extracted object contours are indexed into a database using global and local features as described in the following subsection.

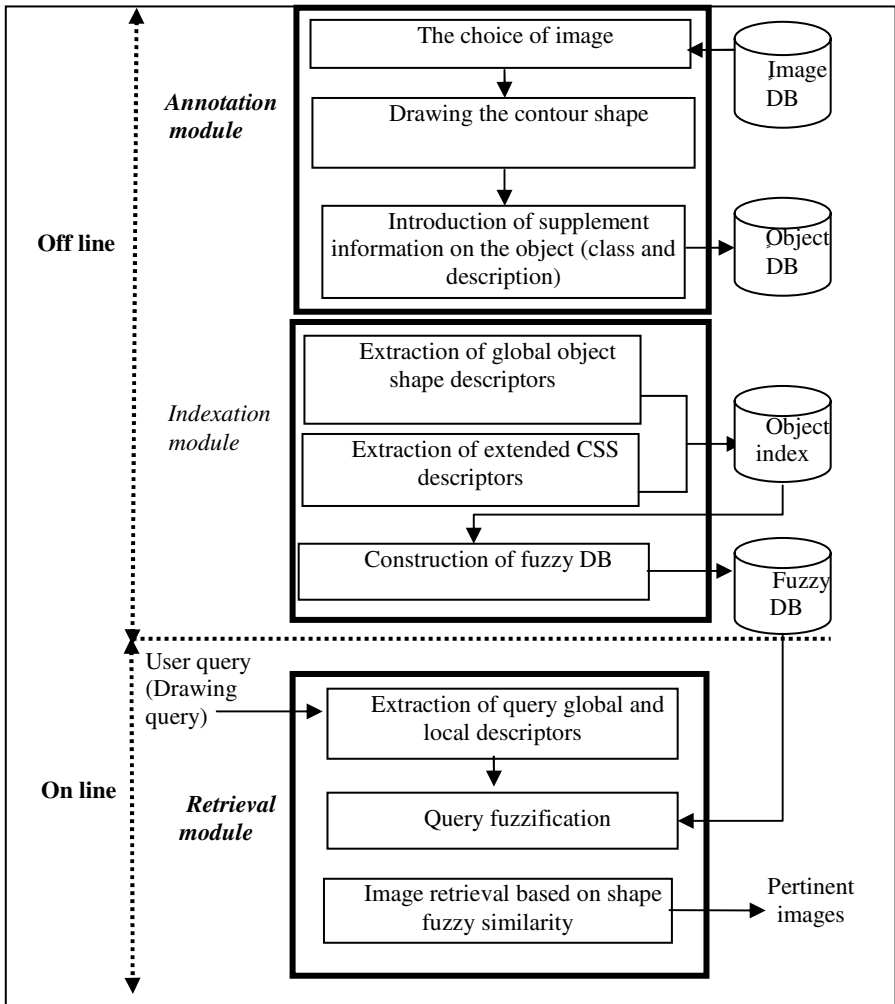


Fig. 2. System architecture

## 2.2 Indexation Module

Indexing an extracted contour can be done using global features, local features, or a combination of the two. Global features typically capture the overall/general shape of a contour, but typically miss all the small local details that may differentiate two very similar contours. They usually have some limitations in modeling perceptual aspects of shapes and perform poorly in the computation of similarity with partially occluded shapes. Local features, on the other hand, capture local variations and/or small details in a contour but may not capture its general shape. Our indexing module uses a mixture of global and local features to remedy both limitations. A good set of features should be invariant to scale, translation, and rotation, and should also be robust, tolerant of noise, and computationally inexpensive.



Fig. 3. The welcome screen of the annotation module graphical user interface

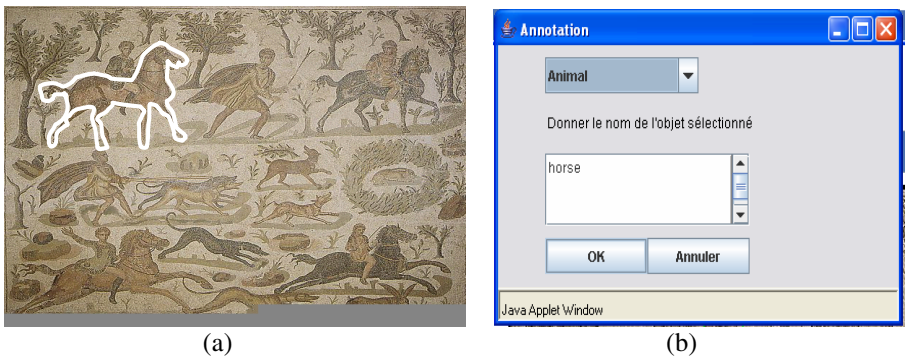


Fig. 4. Object annotation graphical user interface showing (a) a sample object contour extraction and (b) the object semantic textual description

The global features used in our indexing module are circularity and eccentricity. Circularity is a measure of how close a shape is to a circle (which has the minimum circularity measure of  $4\pi$ ). Circularity is a simple and fast feature to compute. It is defined as

$$cir = \frac{P^2}{A} \tag{1}$$

where,  $P$  is the perimeter of the shape and  $A$  is its area. Eccentricity is a global feature that measures how the contour points of a shape are scattered around its centroid. It is defined as

$$ecc = \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \tag{2}$$

where,  $\lambda_{\max}$  and  $\lambda_{\min}$  are the eigenvalues of the matrix  $B$

$$B = \begin{bmatrix} \mu_{2,0} & \mu_{1,1} \\ \mu_{1,1} & \mu_{0,2} \end{bmatrix} \tag{3}$$

and  $\mu_{2,0}$ ,  $\mu_{1,1}$ , and  $\mu_{0,2}$  are the central moments of the shape defined as

$$\mu_{p,q} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \tag{4}$$

with  $\bar{x}$  and  $\bar{y}$  representing the coordinates of the shape’s centroid. Notice that both global features are size, rotation and translation invariant.

The local features we used in our system are an extended version of the curvature scale space (CSS) descriptors introduced by Mokhtarian et al [8, 9]. The CSS descriptors register the concavities of a curve as it goes through successive filtering. The role of filtering is to smooth out the curve and gradually eliminate concavities of increasing size. More precisely, given a form described by its normalized planar contour curve

$$\Gamma(u) = \{(x(u), y(u)) \mid u \in [0,1]\}, \tag{5}$$

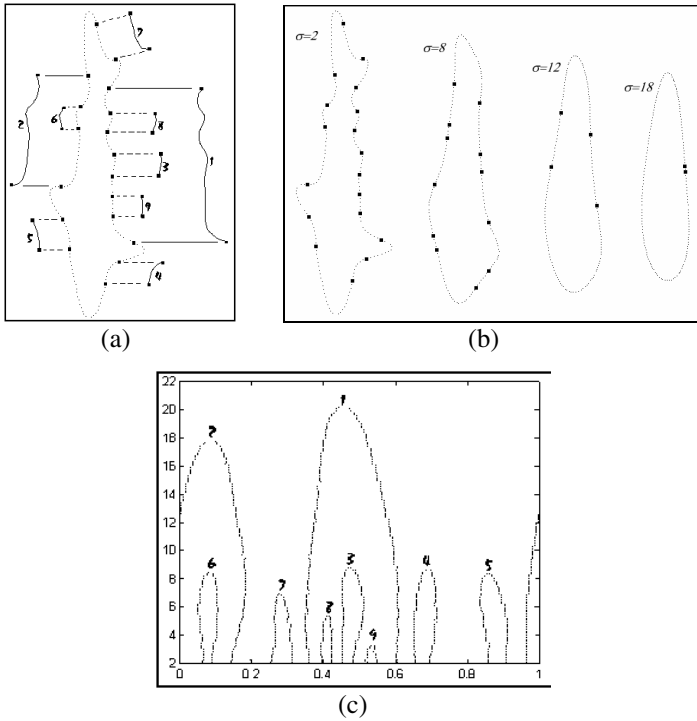
the curvature at any point  $u$  is defined as the tangent angle to the curve and is computed as

$$k(u) = \frac{x_u(u)y_{uu}(u) - x_{uu}(u)y_u(u)}{(x_u(u)^2 + y_u(u)^2)^{\frac{3}{2}}} . \tag{6}$$

To compute its CSS descriptors, a curve is repeatedly smoothed out using a Gaussian kernel  $g(u,\sigma)$ . The main idea behind CSS descriptors is to extract inflection points of a curve at different values of  $\sigma$ . As  $\sigma$  increases, the evolving shape of the curve becomes smoother and we notice a progressive disappearance of the concave parts of the shape until we end up with a completely convex form (Fig. 5). Using the curve’s multi-scale representation, we can locate the points of inflection at each scale (i.e. points where  $k(u,\sigma) = 0$ ). A graph, called CSS image, specifying the location  $u$  of these inflection points vs. the value of  $\sigma$  can be created:

$$I(u, \sigma) = \{(u, \sigma) \mid k(u, \sigma) = 0\} . \tag{7}$$

Figure 5 shows a CSS image of a sample contour. Different peaks present in the CSS image correspond to the major concave segments of the shape. The maxima of the peaks are extracted and used to index the input shape.



**Fig. 5.** Creating the CSS image (c) of a sample contour (a) as it goes through successive filtering (b)

Even though the CSS descriptors have the advantage of being invariant to scale, translation, and rotation, and are shown to be robust and tolerant of noise [8, 9], they are inadequate to represent the convex segments of a shape. Kopf et al [10] presented a solution to remedy this deficiency. The idea they proposed is to create a dual shape of the input shape where all convex segments are transformed into concave segments. The dual shape is created by mirroring the input shape with respect to the circle of minimum radius  $R$  that encloses the original shape (Fig. 6). More precisely, each point  $(x(u),y(u))$  of the original shape is paired with a point  $(x'(u),y'(u))$  of the dual shape such that the distance from  $(x(u),y(u))$  to the circle is the same as that from  $(x'(u),y'(u))$  to the circle. The coordinates of the circle's centre  $O(M_x,M_y)$  are calculated as

$$M_x = \frac{1}{N} \sum_{u=1}^N x(u) \tag{8}$$



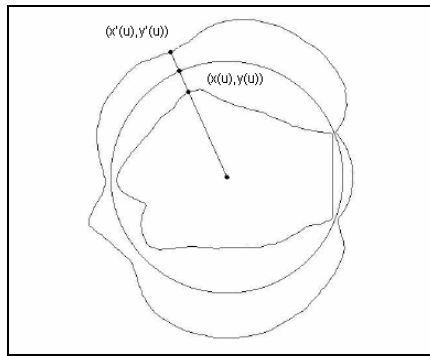
$$M_y = \frac{1}{N} \sum_{u=1}^N y(u). \quad (9)$$

The projected point  $(x'(u), y'(u))$  is located at

$$x'(u) = \frac{2R - D_{x(u),y(y)}}{D_{x(u),y(y)}} (x(u) - M_x) + M_x \quad (10)$$

$$y'(u) = \frac{2R - D_{x(u),y(y)}}{D_{x(u),y(y)}} (y(u) - M_y) + M_y \quad (11)$$

where,  $D_{x(u),y(u)}$  is the distance between the circle's centre and the original shape pixel.



**Fig. 6.** Creating a dual shape with respect to an enclosing circle

The complete set of features (called extended CSS descriptors) we used to index a contour is thus composed of the following:

- Circularity feature (global feature)
- Eccentricity feature (global feature)
- CSS descriptors of original shape (local features)
- CSS descriptors of dual shape (local features)

### 2.3 Image Retrieval Module

Once the contour of an input shape is indexed, a similarity measure is needed to query the database and find similar shapes. Fuzzy similarity measures have been successfully used in many pattern recognition and image processing applications [11, 12, 13, 14]. A fuzzy similarity measure expresses the resemblance degree between two fuzzy sets  $A$  and  $B$  defined as  $A = \{(x, \mu_A(x)) \mid x \in U, \mu_A(x) \in [0, 1]\}$  and  $B = \{(x, \mu_B(x)) \mid x \in U, \mu_B(x) \in [0, 1]\}$  where,  $U = \{x_1, x_2, \dots, x_n\}$  is a discourse universe and  $\mu_A$  and  $\mu_B$  are the membership functions of  $A$  and  $B$ , respectively. We chose to use Lukasiewicz fuzzy similarity measure [15] in our image retrieval module. It is defined as

$$s = \inf (1 - |\mu_A(x) - \mu_B(x)|) \tag{12}$$

The value of this similarity measure is highest (i.e. 1) when we have a perfect match. To apply this similarity measure, we first need to represent each feature by a suitable membership function. Figure 7, for example, shows the membership functions we used with the circularity feature. Each feature value is thus fuzzified by their membership degree to the corresponding membership functions. When a user enters a query into the retrieval module, the query features are fuzzified before being compared to entries in the database. We set up our retrieval system so that it returns all database images that have a similarity measure of 0.5 or higher with respect to the query shape.

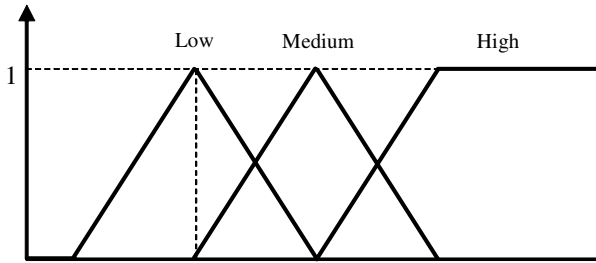


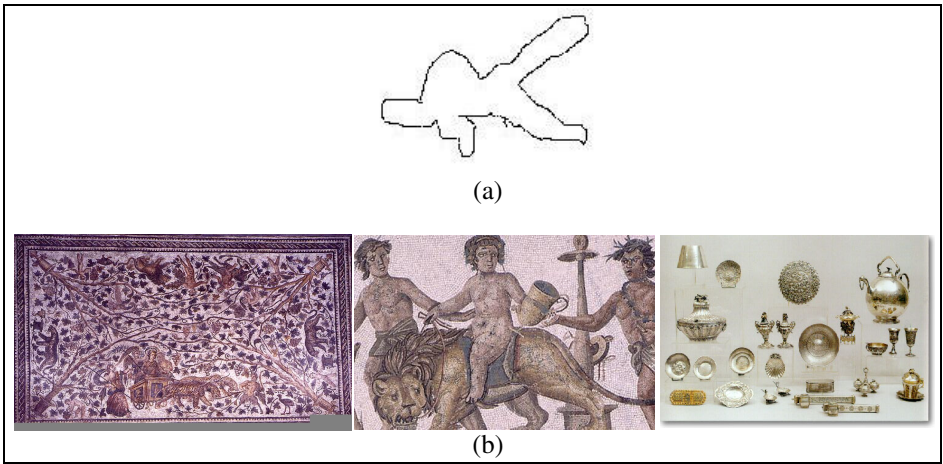
Fig. 7. Circularity membership functions

### 3 System Evaluation

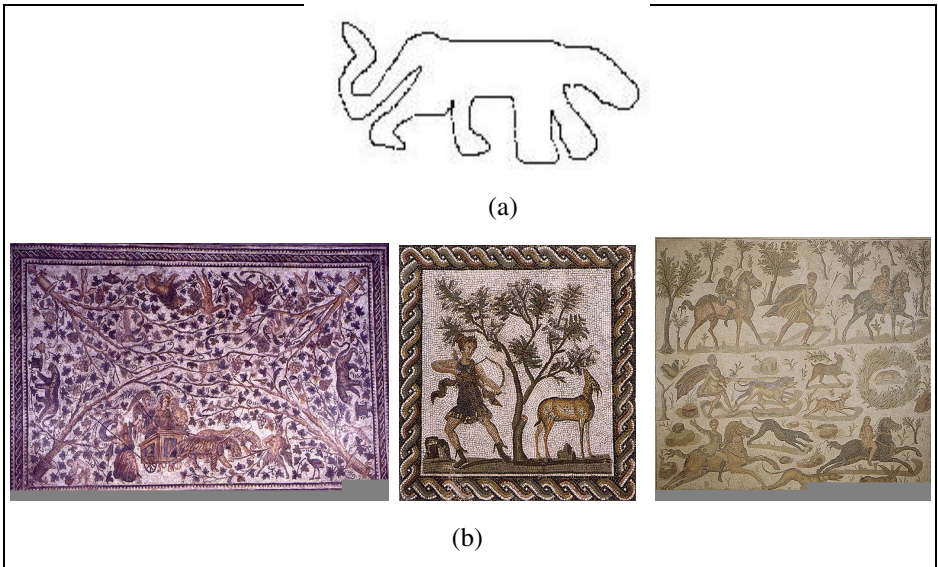
The image database we used in the testing of our system consists of color and grey scale images of historic documents, artifacts, and mosaics dating as far back as the second century CE. These images are provided to us by the National library of Tunisia, the National archive of Tunisia, and a collection of national Tunisian museums. The images are manually annotated and automatically indexed as described in previous sections. We built a graphical user interface where a user can draw with a computer stylus and pad a sketch of the shape he/she is looking for. The user also has the option to specify the class of the shape he/she is looking for (e.g. person, animal, etc.). The system indexes the user’s query and matches it to the images in the database using the fuzzy similarity measure described in the previous section. The system returns only images that have a similarity measure greater or equal 0.5. Table 1 shows the system’s precision and recall rates for three specific object classes (person, animal, and

Table 1. System’s recall and precision rates

Object class	Recall rate (%)	Precision rate (%)
Person	85.7	60.0
Animal	83.0	45.5
Graphics	44.4	88.9
Overall average	83.0	60.0



**Fig. 8.** (a) Shape query loosely resembling a person and (b) the first three matching images



**Fig. 9.** (a) Shape query resembling the outline of an animal and (b) the first three matching images

graphics) and the overall average rates for all classes in the database. Given the complexity of the images in the database, the overall recall and precision rates are very respectable. Figures 8 and 9 show some sample queries and the top 3 image matches returned by the system. Notice that, even though some queries are at an orientation different than that of the database images and some are heavily distorted (Fig. 8), the system was still able to retrieve very pertinent images.

## 4 Conclusion

We designed a system to index and retrieve images of historic documents and artifacts. The system uses extended CSS descriptors to index shape contours and a fuzzy similarity measure to retrieve pertinent shapes in the database. The overall recall and precision rates were 83% and 60%, respectively. Future work includes adding more features and trying other similarity measures to improve both recall and precision rates.

## Acknowledgements

The authors would like to acknowledge the financial support of this work by grants from the General Direction of Scientific Research and Technological Renovation (DGRSRT), Tunisia, under the ARUB program 01/UR/11/02. Part of this research was also funded by the Tunisian-Egyptian project "Digitization and Valorization of Arabic Cultural Patrimony". The authors would like to thank the National Library of Tunisia for giving them access to its large image database of historic Arabic documents.

## References

1. National Library of Tunisia, <http://www.bibliotheque.nat.tn>
2. National Archive of Tunisia, <http://www.archives.tn>
3. Maghrebi, W., Khabou, M.A., Alimi, A.M.: A System for Indexing and Retrieving Historical Arabic Documents Based on Fourier Descriptors. In: International Conference on Machine Intelligence ACIDCA - ICMI '2005, Tozeur, Tunisia, pp. 701–704 (2005)
4. Zaghden, N., Charfi, M., Alimi, A.M.: Optical Font Recognition Based on Global Texture Analysis. In: International Conference on Machine Intelligence, Tozeur, Tunisia, pp. 712–717 (2005)
5. Boussellaa, W., Zahour, A., Alimi, A.M.: A Methodology for the Separation of Foreground/Background in Arabic Historical Manuscripts using Hybrid Methods. In: 22nd Annual Symposium on Applied Computing, SAC 2007, Document Engineering Track, Seoul, Korea (2007)
6. Schomaker, L., Vuurpijl, L., Deleau, E.: New Use For The Pen: Outline-Based Image Queries. In: 5th International Conference on Document Analysis and Recognition, Piscataway, NJ, pp. 293–296 (1999)
7. Vuurpijl, L., Shomaker, L., Broek, E.: Vind(x): Using the User Through Cooperative Annotation. In: 8th International Workshop on Frontiers in Handwriting Recognition, Canada, pp. 221–225 (2002)
8. Mokhtarian, F., Abbasi, S., Kittler, J.: Efficient And Robust Retrieval By Shape Through Curvature Scale Space. In: 1st International Workshop on Image Databases and Multimedia Search, pp. 35–42 (August 1996)
9. Mokhtarian, F., Abbasi, S., Kittler, J.: Robust And Efficient Shape Indexing Through Curvature Scale Space. In: British Machine Vision Conference, pp. 53–62 (1996)
10. Kopf, S., Haenselmann, T., Effelsberg, W.: Shape-Based Posture Recognition In Videos. Proceedings of Electronic Image 5682, 114–124 (2005)

11. Wang, X.Z., Baets, B.D., Kerre, E.: A Comparative Study of Similarity Measures. *Fuzzy sets and systems* 73, 259–268 (1995)
12. Wang, W.-J.: New Similarity Measures on Fuzzy Sets And On Elements. *Fuzzy sets and systems* 85, 305–309 (1997)
13. Mitchell, H.B.: On the Dengfeng-Chuntian Similarity Measure and Its Application To Pattern Recognition. *Pattern Recognition Letters* 24, 3101–3104 (2003)
14. Van Weken, D., Nachtegaele, M., Witte, V., De Schulte, S., Kerre, E.E.: A Survey on The Use And The Construction Of Fuzzy Similarity Measures In Image Processing. In: *IEEE International Conference on Computational Intelligence for Measurement Systems and Application*, pp. 187–191. Giardini Naxos, Italy (2005)
15. Ben Ayed, D., Ellouze, N.: Classification Phonétique Basée Sur Les Mesures De Similarités Floue: Application Aux Signaux De Parole De Type Voyelle. In: *JTEA Conference, Sousse Nord, Tunisia* (2002)

# PCB Inspection Using Image Processing and Wavelet Transform

Joaquín Santoyo<sup>1</sup>, J. Carlos Pedraza<sup>2</sup>, L. Felipe Mejía<sup>1</sup>, and Alejandro Santoyo<sup>3</sup>

<sup>1</sup> Universidad Tecnológica de Querétaro, Av. Pie de la Cuesta s/n Lomas de San Pedrito  
Peñuelas Querétaro, Qro. C.P. 76148 México  
jsantoyo@uteq.edu.mx

<sup>2</sup> Centro de Ingeniería y Desarrollo Industrial, Av. Pie de la Cuesta No. 702 Desarrollo san  
Pablo, Querétaro, Qro. C.P. 76130 México  
jpedraza@cidesi.mx

<sup>3</sup> Universidad Autónoma de Querétaro, Centro Universitario, Cerro de las Campanas  
Querétaro, Qro. C.P. 76010 México  
alex@uaq.mx

**Abstract.** In electronics mass-production manufacturing, printed circuit board (PCB) inspection is a time consuming task. Manual inspection does not guarantee that PCB defects can be detected. In this paper, a spatial filtering and wavelet-based automatic optical inspection system for detect PCB defects is presented. This approach combines wavelet image compression utility and spatial filtering. Defects are detected by subtracting the approximations of reference image wavelet transform and test image wavelet transform followed by a median filter stage. Finally, defect image is obtained by computing the inverse wavelet transform. Advantages of this approach are also described.

## 1 Introduction

Manual inspection of printed circuit boards (PCB's) has been displaced by automatic optical inspection due to human limitations. Automatic optical inspection (AOI) do the same labor as manual inspection with great advantages: it can recognize line width errors, shorts, pinholes, etc. Automatic inspection also has advantages over electrical contact testing: it can detect possible electric leakages, inductance and capacitance parasite due to nicks, mousebite etc.

Inspection methods can be grouped in three main categories, as stated by Moganti *et al.* [1]: referential methods, rule-based methods and hybrid methods. Referential method needs a "good" sample to compare point-to-point with a regular sample. It has the disadvantage that some process deviations could be interpreted as defects.

Non-referential methods or sometimes called design-rule verification method, do not require a gold sample to work with; they work with the PCB specifications design as a set of rules.

Finally hybrid methods combine the advantages of referential and rule-base methods. Unfortunately this approach is a time consuming method due to double check.

Wavelet transform has been used in vision field for different applications: fingerprint recognition [2], medical image analysis [3], image noise removal [4], image compression [5], and others.

Although some work on printed circuit board verification have been developed in recent years [6], there are some aspects like image denoising that can be explored using other techniques.

The aim of this work is find defects in printed circuit boards combining the recent wavelet transform for improve data processing and spatial filtering to reduce or eliminate noise in a proper way.

## 2 Wavelets

The continuous Wavelet Transform  $f(a,b)$  of a continuous signal  $x(t)$  is defined [7]:

$$f(a,b) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right) dt. \quad (1)$$

To assure the perfect reconstruction of the original signal, the wavelet Transform. must satisfy

$$C_{\Psi} = \int_{-\infty}^{\infty} \frac{|\Psi'(\omega)|^2}{\omega} d\omega < \infty \quad (2)$$

where  $\Psi'(\omega)$  denotes wavelet Fourier transform. This condition is the so-called *admissibility criterion* for the wavelet  $\psi(t)$ .

The continuous wavelet transform is difficult to evaluate for all scales and positions. Using the Discrete Wavelet Transform a suitable set of scales and positions can be chosen.

The discrete wavelet transform of an image or function  $f(x, y)$  of size  $M \times N$  is

$$W_{\varphi}(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \varphi_{j_0, m, n}(x, y) \quad (3)$$

$$W^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \psi^i_{j, m, n}(x, y) \quad (4)$$

Where  $i = \{H, V, D\}$

## 3 Algorithm Proposal

This proposal is a model-based method due to the comparison of a reference image and a test image. A second level Haar wavelet is applied to reference image. Haar

wavelet was chosen due to this wavelet is simplest, fastest and suitable for this approach [8]. The wavelet output approximations (*coef1*) are stored in memory and this step is made just once. A second level Haar wavelet is applied to test image and then the output wavelet transform (*coef2*) is subtracted from memory. The resulting matrix (*coef3*) has the information of defects if any and some noise. Noise is eliminated thresholding absolute values of *coef3* and then applying a 5 x 5 median filter. This algorithm can be resumed as follows (Fig 1):

- $coef1(x,y)$  Second level approximation Haar wavelet transform of reference image.
- $coef2(x,y)$  Second level approximation Haar wavelet transform of test image.
- $coef3(x,y) = coef1(x,y) - coef2(x,y)$
- if  $coef3(x,y) > t$  then  $coef3(x,y)=1$ , if not,  $coef3(x,y)=0$ .
- $Coef4(x,y) = Defecto(m,n) =$  inverse wavelet transform of  $coef4(x,y)$

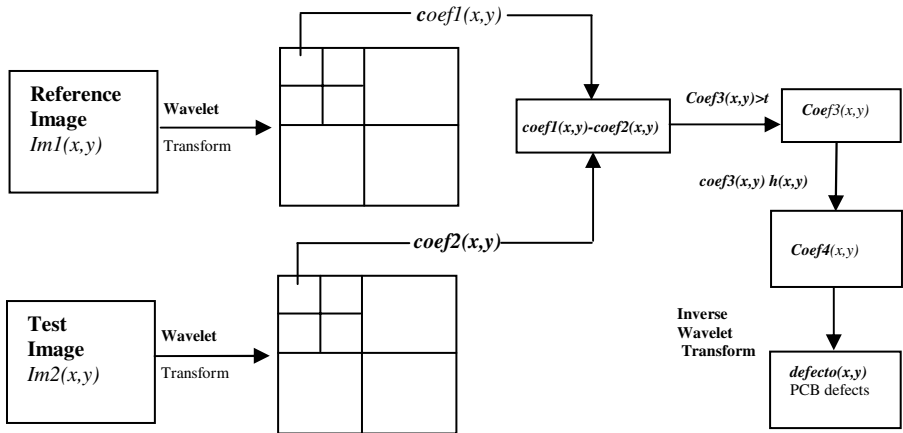
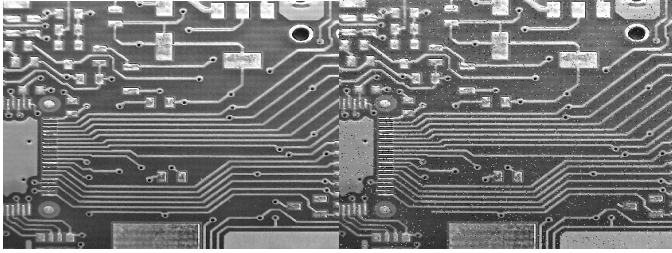


Fig. 1. Defect detection algorithm

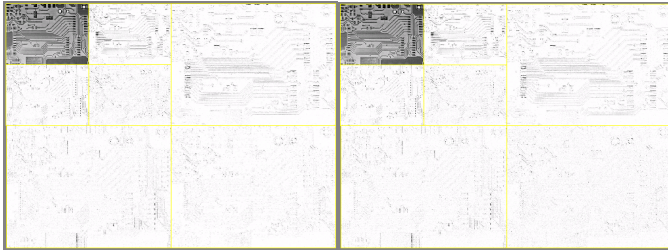
## 4 Experiments and Results

Two images of size 1000 x 1000 were used for testing this approach. Reference and test images are gray scale images of a real printed circuit board. Test image is a defective PCB corrupted with salt & pepper noise (Fig 2). The alignment of the test image is made manually. The approximation part of second level Haar wavelet for the reference image and test image is calculated. The size of the resulting approximation matrix is 250 x 250. Only the approximation part of the second level wavelet transform is used due to the great advantage of having all the meaningful information of the original image in just 1/4 of the original size.



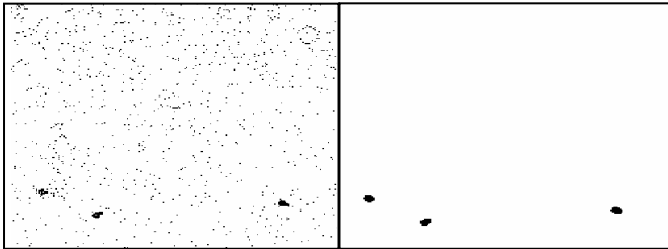


**Fig. 2.** a) Reference Image b) Test image



**Fig. 3.** Approximations for a) reference image b) test image

For this implementation was found that 6 was the best threshold value before filtering. Several window filter sizes were used to eliminate the image noise; a  $5 \times 5$  window size gave the best results. Finally second level Haar wavelet inverse transform is applied to  $coef3(x,y)$  and it is obtained an image that contains the detected defects (fig 4).



**Fig. 4.** a) Difference image with noise b) Defects detected after filtering

A classical image processing approach to detect defects was implemented at same time to compare results with the proposed algorithm. The stages for this approach were binarization, image difference and filtering all in full resolution. Defects were found without problem. Table 1 shows the comparison of operation time for every step for both implementations. Using the Wavelet-based approach a time reduction about 50% is obtained. The time consuming task is filtering. In both approaches,

filtering is used but in different domain: Wavelet-based (1/4 of original size) and full resolution. Due to this data compression, a reduction time in data processing can be obtained.

**Table 1.** Inspection time for a) classical image processing approach b) wavelet-based approach

<b>Operation time (s)</b>		
<b>Operation</b>	<b>Classical approach</b>	<b>Wavelet-based</b>
Read image	0.546	0.546
Second wavelet transform	Not used	0.094
Binarization	0.078	Not used
Image difference	0.047	0.001
Median filtering	0.813	0.094
Inverse wavelet	Not used	0.026
Total	1.484	0.761

Matlab was chosen as a platform to develop and test the algorithm using the Matlab's Wavelets Toolbox. Tests were implemented in a laptop with a Centrino Duo processor.

## 5 Conclusions and Future Work

The wavelet transform approximation is a compressed image that preserves the meaningful information necessary to complete the image processing and be able to detect any defect.

The combination of using wavelets transforms and spatial filtering used in this approach has shown that defects in a printed circuit board can be detected in a similar way as a classical image processing method can do. This represents a great advantage when the verification of PCB's is in real time due to reduction of processing data. Future work will be directed towards the classification of the defects not just found them. Aspects relative to defects like size, position, orientation and classification techniques must be investigated

## References

1. Moganti, M., Ercal, F., Dagli, C.H., Tsunekawa, S.: Automatic PCB Inspection Algorithms: A Survey. *Computer Vision and Image Understanding* 6,3(2), 287–313 (1996)
2. Tico, M., Immonen, E., Ramo, P., Kuosmanen, P., Saarinen, J.: Fingerprint recognition using wavelet features. In: *International Symposium on Circuits and Systems*, vol. 2(6-9), pp. 21–24 (2001)
3. Zhang, X., Yang, Y., Xu, X., Zhang, M.: Wavelet Based Neuro-Fuzzy Classification for EMG Control. In: *International Conference on Communications, Circuits and Systems*, vol. 2, pp. 1087–1089 (2002)

4. Huang, X., Madoc, A.C., Cheetham, A.D.: Multi-Noise Removal from Images by Wavelet-Based Bayesian Estimator. In: International Symposium on Multimedia Software Engineering, vol. 13(15), pp. 258–264 (2004)
5. Lazar, D., Averbuch, A.: Wavelet video compression using region based motion estimation and compensation. In: International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 1597–1600 (2001)
6. Ibrahimt, Z., Al-Attas, S.A.R., Osamu O., Mokji, M.M.: A Noise Elimination Procedure for Wavelet-Based Printed Circuit Board Inspection System. In: 5th Asian Control Conference, pp. 875–880
7. Mertins, A.: Wavelets, Filter Banks, Time-Frequency Transforms and Applications, 1st edn. John Willey & Sons, England (1999)
8. Chengjiang, L.: Dissertation: Time and Space Efficient Wavelet Transform for Real-Time Applications. The Ohio State University (1999)

# A Single-Frame Super-Resolution Innovative Approach

Luz A. Torres-Méndez<sup>1</sup>, Marco I. Ramírez-Sosa Morán<sup>2</sup>, and Mario Castelán<sup>1</sup>

<sup>1</sup> CINVESTAV Campus Saltillo, Ramos Arizpe, Coahuila, 25900, México  
`abril.torres@cinvestav.edu.mx`

<sup>2</sup> ITESM Campus Saltillo, Saltillo, Coahuila, 25270, México

**Abstract.** Super-resolution refers to the process of obtaining a high resolution image from one or more low resolution images. In this work, we present a novel method for the super-resolution problem for the limited case, where only one image of low resolution is given as an input. The proposed method is based on statistical learning for inferring the high frequencies regions which helps to distinguish a high resolution image from a low resolution one. These inferences are obtained from the correlation between regions of low and high resolution that come exclusively from the image to be super-resolved, in term of small neighborhoods. The Markov random fields are used as a model to capture the local statistics of high and low resolution data when they are analyzed at different scales and resolutions. Experimental results show the viability of the method.

## 1 Introduction

Image super-resolution refers to the process of obtaining a high resolution (HR) image from one or more low resolution (LR) images. The resolution of an image is lost due to different factors such as limited hardware, noise, scaling, defocus, etc. The immediate form of augmenting the resolution of an image is by using sensors of greater resolution, i.e. having a greater number of small-size photo-sensors. This, however, implies a greater cost. Another, more economic form of augmenting the resolution of images is using algorithms especially design for this task, which apart from obtaining images with a resolution similar to that obtained from a high quality sensor, they must be fast and robust.

In the literature we can find several methods for the super-resolution problem, in most of them the input is a set of LR images from which a HR image is obtained. For the limited case, where the only input is the LR image to be super-resolved, is almost impossible to recover high-frequency information accurately. However, in the last five years, methods have been proposed that base their learning stage in the correlation between regions with low and high resolution of images that are similar to the one to be improved [5].

In this research, we propose to capture the intrinsic characteristic in regions containing high frequency pixels which give the appearance of high definition in the HR images. The proposed method is based on a statistical learning that uses exclusively the input LR image, independently of its content and nature.

We have developed a multi-scale framework that uses various levels of scale (size of the image) at different resolutions to extract fundamental characteristics between regions of low and high resolution. The high frequencies of the image of the lowest level of scale, or the smallest size image, can be considered at least in appearance to the human eye, as an HR image. By using the corresponding LR image at that level, we can build at each level the training set, which is composed of small neighborhoods to estimate the HR image in the upper level and successively.

We describe the characteristics of the input image to multiple levels of scale in order to provide locality and efficiency in the statistical analysis. The inter-level relationships of adjacent scales and between intra-scale neighbors make Markov models be the natural selection. In this framework, a series of probabilistic models are learned, each of them can be effectively characterized by a reasonable small number of parameters and then estimated with a limited number of training cases. These ideas are illustrated with a variety of examples of images under different scenarios.

## 2 Related Work

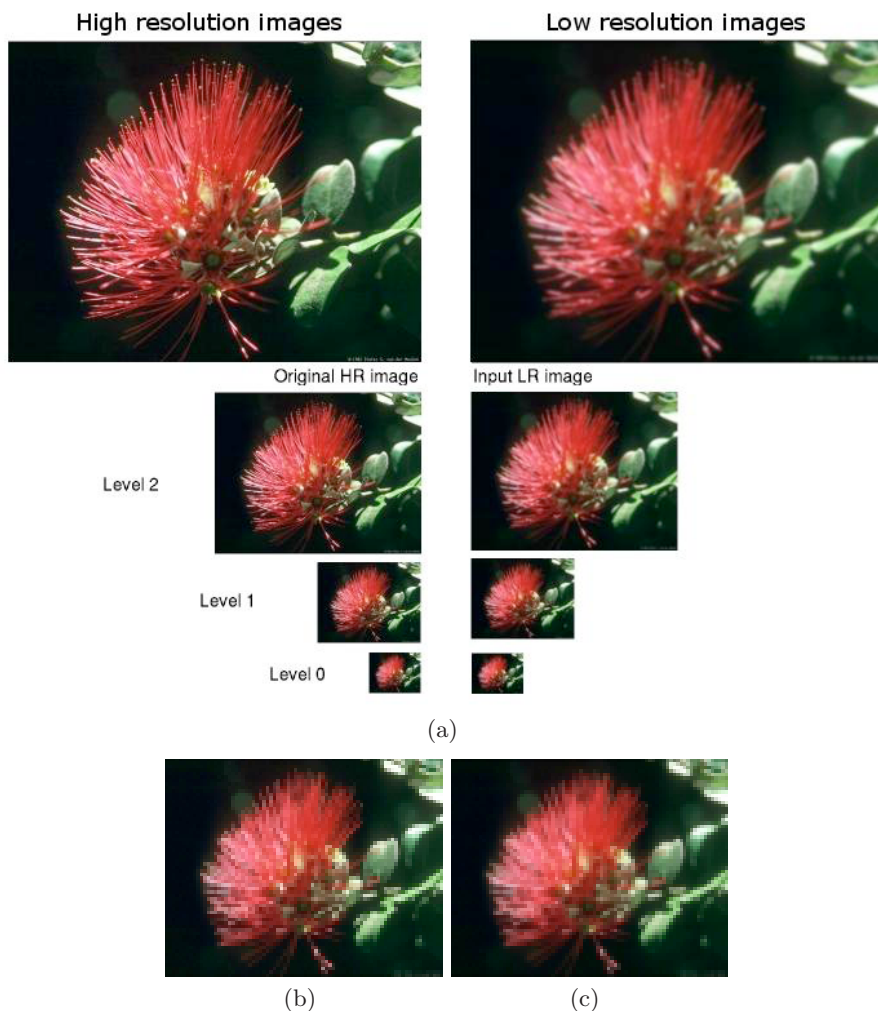
The first to propose the idea of super-resolution was Tsai and Huang. They used the frequency domain in their approach [15]. Kim et al. [10] use a recursive algorithm, also in the frequency domain, for the restoration of the super-resolution of images from blurry and noisy observations. Many of the research work in image analysis for the extraction of characteristics and geometry are based on probabilistic methods. For the problem of super-resolution, in [8], the authors use the method of the maximum a posteriori (MAP) to estimate the register parameters and the HR image for various observations simultaneously. Other approaches include techniques of super-resolution based on MAP with Markov Random Fields (MRF) using the blurry characteristics as clue. Chiang and Boulton [4] use edge models and a blurry local estimate to develop an algorithm for super-resolution based on edges. To reconstruct the HR image they also apply a deformation which is based in the concept of integrated resampling that deforms the image subject to certain constraints. Other interesting work is that of Candocia and Principe [2], who attack the super-resolution problem assuming that the correlated neighbors stay similar in all scales, and this *a priori* information is learned locally from the available sampling images through all the scales. When a new image is presented, a core is automatically selected such that achieves the best reconstruction of each local region and the resulted HR image is reconstructed using a simple convolution operation. The use of SVM learning algorithm for regression is presented in [11]. Such technique approaches the super-resolution problem as an estimation problem with a criterion of image correctness and it is improved adding a structure in the DCT coefficients. The proposed algorithm requires information learned from training data in addition to observed information. In [9], a learning-based approach that outperforms standard interpolation and wavelet-based learning techniques is proposed. The contourlet learning method proposed is an extension of the Cartesian wavelet

transform in two dimensions using multiscale and directional filter banks which makes it capable of capturing the geometrical smoothness of the contour along any possible direction. In this result the use of a database of HR images is still needed. If the single frame is a LR image formed of patches with overlaps, the problem can be reduced to obtain a HR image corresponding to only one of the patches, in such case it should be assumed that the HR patches related to the rest of LR patches are known and such pairs are used as training information. The problem presented in [14] is the idea of a multi-frame converted into a single frame using the patch-image approach. In that paper a solution to find the nearest neighbors to the patch being analyzed is proposed. The one-pass example-based algorithm is presented in [5] requiring also a nearest-neighbor search in a training set for a vector derived from each patch of local image data, this represented a first attempt to achieve resolution independence in image-based representation. Learning-based methods represent some of the current approaches to single-frame super-resolution. Such efforts are not attempting to remove the aliasing or the noise/blurring present in the observed data which is still an open problem when using a single image to obtain the HR image. Some examples of the last decade proposing solutions to this very ill-posed problem are discussed in [11,12,13].

### 3 Our Framework

In this section, a methodology that describes relevant HR characteristics from one low resolution image is presented. A multi-scale scheme of the same image is used to learn the high frequencies in order to obtain a HR image. The super-resolution problem based on multiple scales can be seen as a restoration problem. The first idea in which our approach is based is that an image can be modeled as a sampling function of a stochastic process based on the Gibbs distribution, i.e., as a Markov Random Field (MRF) [7]. The task of augmenting the resolution to a low resolution image is considered then as a process of assigning intensity values to each pixel of the input image that best describe its high frequency regions.

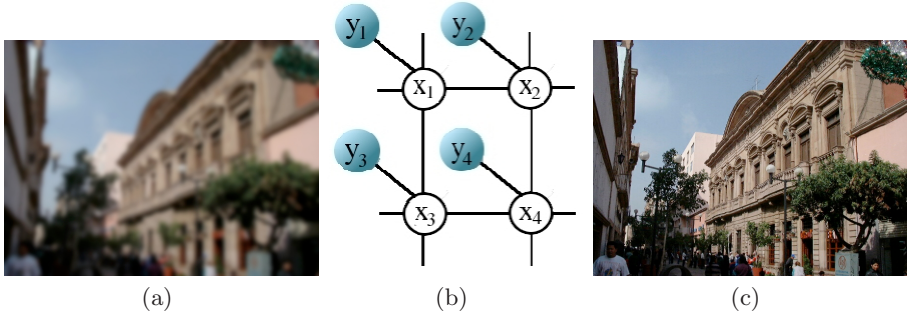
Figure 1 illustrates the general idea of our approach. The left column shows the HR images at different sizes and the right column shows the corresponding LR images. It can be seen that as the images become smaller, the images and therefore their resolutions, tend to be very similar. Since the HR version of the image is unknown, we can exploit this fact, and assume that the HR version of the LR image at level 0 (the smallest image) are identical (see the zoomed versions in Figure 1(c),(d)). Thus, we can start using their high frequency features from a set of small neighborhoods to achieve an inter-level training/learning process. In other words, the information of all high frequency regions coming from lower levels is to be propagated and preserved in all levels. The MRF model has the ability of capturing the characteristics between the training sets of each level and then use these characteristics for learning a marginal probability distribution that will be used in the images of upper levels up to the desired level. The power of our technique resides in that only a small set of training neighborhoods



**Fig. 1.** Multi-scaled super-resolution approach. (a) Left column shows the HR images and the right column shows the corresponding LR images. The last row (level 0) contains the smallest images, and it can be seen that both, the HR and LR images, (b) and (c), look very similar to the human eye. This LR image (c) represents our starting point to learn the high-frequencies, train our statistical model and propagate the information to upper levels. Please refer to the text for detailed description.

coming from the same input image at different sizes (levels) is required to make the super-resolution. Each pair in the training set of each level is composed by a neighborhood of low resolution with its corresponding high resolution neighborhood. The statistical relationships are learned directly from the training data, without having to consider the type or nature of the image.





**Fig. 2.** (b) Pairwise Markov network at level  $l$  used to model the joint probability distribution of the system. Observations nodes,  $y$  represent a LR image patch from (a), and hidden nodes  $x$ , a HR image patch in image (c) to be inferred.

### 3.1 The MRF Model

Given a monochromatic LR image  $\mathbf{P}$  of size  $h_p \times w_p$  pixels. We want to estimate the HR image  $\mathbf{S}$  of size  $h_s \times w_s$ , of equal or greater size of the input image  $\mathbf{P}$ . From  $\mathbf{P}$ , we generate  $L$  images of smaller size (scaled), that we call observable images  $l_1, l = 0, \dots, L$ . The image in the lowest level ( $I_0$ ), corresponds to the smallest image. Here, we consider (and take advantage of) that the image  $I_0$  contains relevant high resolution characteristics. Our hypothesis comes from this fact and also from the observation that the human eye can recognize more easily these characteristics in smaller images of low resolution than in bigger images of low resolution. A MRF model (also known as a Markov net) is defined as a set of hidden nodes  $x_i$  (white circles in the graph of Figure 2) representing local neighborhoods in the output image  $\mathbf{S}$ , and the observable nodes  $y_i$  (shaded circles in the graph) representing local neighborhoods of the input image  $\mathbf{P}$ . Each local neighborhood is defined by a central pixel  $i$  of the respective images. At each level, we construct a Markov net, and as the images are decreasing in size, the size of their corresponding local neighborhoods must also vary. For example, for the smallest image (level 0) the size of the neighborhood may be of  $5 \times 5$  pixels, for level 1,  $7 \times 7$  pixels, level 2,  $9 \times 9$  pixels, and so on.

Denoting the pairwise potentials between the variables  $x_i$  and  $x_j$  by  $\Psi_{ij}$  and the local evidence potentials associated with the variables  $x_i$  and  $y_i$  by  $\Phi_i$ , the joint probability of the MRF model under the variable instantiation  $x = (x_1, \dots, x_m)$  and  $y = (y_1, \dots, y_m)$ , can be written [7] as:

$$P(x, y) = \frac{1}{Z} \prod_{ij} \psi_{ij}(x_i, x_j) \prod_i \phi_i(x_i, y_i), \tag{1}$$

where  $Z$  is a normalization constant. We wish to maximize  $P(x, y)$ , i.e., we want to find the most likely state for all hidden nodes  $x_i$ , given all the evidence nodes  $y_i$ . At each level, as the information coming from both type of nodes observes only a small vicinity, such local information can be ambiguous. Propagating



information between regions can solve this ambiguity. The compatibility functions allow to establish high compatibilities (o low) to neighboring pixels according to the particular application. In our case, it is important to preserve discontinuities (edges) in the input image as these represent the high frequencies and we must avoid to smooth them in the output image. Then, we assign a high compatibility between the neighboring pixels that have similar intensity values and low compatibility between neighboring pixels that present abrupt changes in their intensity values. These potentials are used in messages that are propagated between pixels to indicate the intensity combinations that each image pixel must have. The value of a pixel in  $S_L$  is synthesized estimating the maximum *a posteriori* solution (MAP) of the MRF model by using the training set  $T_L$ , which would correspond to the complete Markov net obtained from one level down. The MAP solution of the MRF model is:

$$\mathbf{x}_{MAP} = \arg \max_{\mathbf{x}} P(\mathbf{x} | \mathbf{y}), \tag{2}$$

where

$$P(\mathbf{x} | \mathbf{y}) \propto P(\mathbf{y} | \mathbf{x})P(\mathbf{x}) \propto \prod_i \phi_i(x_i, y_i) \prod_{(i,j)} \psi_{ij}(x_i, x_j) \tag{3}$$

Computing the conditional probabilities in an explicit form to infer the exact MAP solution en the MRF models is intractable. We cannot efficiently represent or determine all the possible combinations between pixels with its associated neighborhoods. Various techniques exist for approximating the MAP estimate, such that the Markov Chain Monte Carlo (MCMC), iterated conditional modes (ICM), maximizer or posterior marginals (MPM), etc. In this work, we calculate a MAP estimate based on the learning scheme of the Harkov network, and using the Belief Propagation algorithm (BP), as proposed by Freeman *et al.* [6].

The compatibility functions  $\phi(x_i, y_i)$  and  $\psi(x_i, x_j)$  of each level are learned from the respective sets using the method based on neighborhoods. It is assumed that these functions obey a Gaussian distribution to model Gaussian noise. The  $\phi_i(x_i, y_i)$  compatibility function is defined as follows:

$$\phi_i(x_i, y_i) = e^{-|y_i - y_{x_i}|^2 / 2\sigma_i^2} \tag{4}$$

where  $x_i$  is a candidate region containing characteristics of high resolution,  $y_{x_i}$  is the low resolution corresponding region of  $x_i$ , and  $y_i$  is the region in the image of the current level being analyzed. In each level, the low resolution image is divided such that the neighborhoods or corresponding regions of high resolution overlap. If the overlapping pixels of two nodes are similar, then the compatibility between those states is high. We define  $\psi(x_i, x_j)$  as:

$$\psi_{ij}(x_i, x_j) = e^{-d_{ij}(x_i, x_j) / 2\sigma_i^2} \tag{5}$$

where  $d_{ij}$  is the difference between the neighborhoods of pixels  $i$  and  $j$ . The training set of the level  $l$  is composed by pairs of small regions of low resolution with its corresponding high resolution information both coming from the lower level  $l - 1$ . Thus, the compatibility functions depend directly on the level in which the estimation is to be done.

### 3.2 The MRF-MAP Inference Using BP

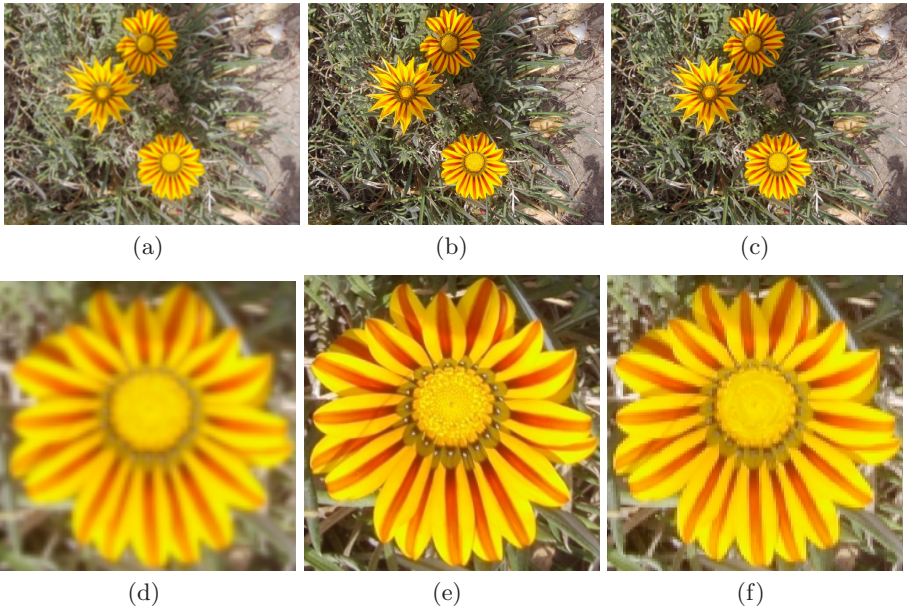
Belief propagation (BP) was originally introduced as an exact algorithm for tree-structured models [13], but it can also be applied for graphs with loops, in which case it becomes an approximate algorithm, leading often to good approximate and tractable solutions [16]. For MRFs, the belief propagation is an inference method for estimating efficiently the Bayesian belief in the network by passing messages in an iterative way between the neighboring nodes. The message sent from node  $i$  to any of its adjacent nodes  $j$  is:

$$m_{ij}(x_j) = Z \sum_{x_i} \psi(x_i, x_j) \phi(x_i, y_i) \prod_{k \in \mathcal{N}(i) \setminus \{j\}} m_{ki}(x_i) \quad (6)$$

where  $Z$  is the normalization constant. The maximum a posteriori scene patch for node  $i$  is:

$$x_{iMAP} = \arg \max_{\mathbf{x}_i} \phi(x_i, y_i) \prod_{j \in \mathcal{N}(i)} m_{ji}(x_i). \quad (7)$$

In our experiments, the algorithm usually converges in less than 10 iterations for the top level and less than 5 for the bottom level. And it is also important to note that the belief propagation algorithm is faster than many traditional inference methods. The candidates for each neighborhood are taken from the training set. For each neighborhood of low resolution, we search in the training set for neighborhoods that best resemble the input. The neighborhoods of high

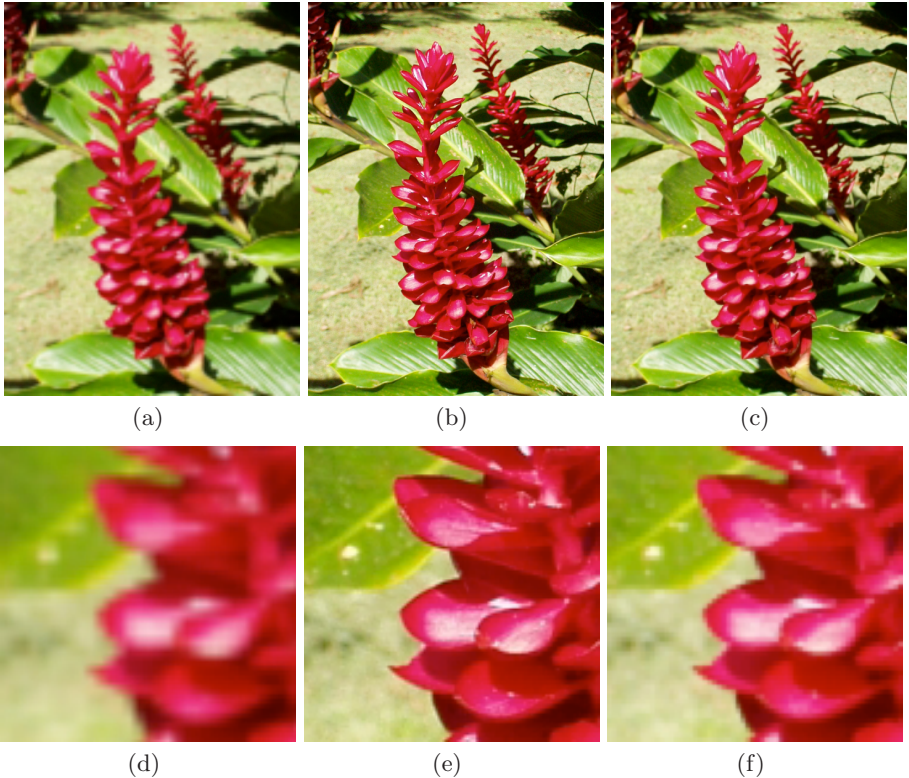


**Fig. 3.** (a,d) Input LR image. (b,e) Original HR image. (c,f) The synthesized HR image. (d-f) are the image regions in actual size from (a-c), respectively.

resolution corresponding to the  $k$  best neighborhoods are used as possible states for the hidden nodes.

## 4 Experimental Results

We test the proposed method in a variety of images containing different objects and scenes. These images were taken with a commercial digital camera. In order to evaluate our method, each scene was taken twice, using two sizes and resolutions, one HR image of  $1024 \times 768$  pixels and the other of  $640 \times 480$  pixels of medium resolution. This picture was scaled to the size of the first one to obtain an LR image of  $1024 \times 768$ , which is given as an input to our super-resolution algorithm. The size of the neighborhoods in all the experiments was set to  $7 \times 7$  pixels to the top level,  $5 \times 5$  to the intermediate level and  $3 \times 3$  to the bottom level. The number of possible candidates  $k$ , was fixed to 10 and the number of levels  $L$  to 3. Figure 3a shows an example of an input LR image. Figure b depicts the original HR image and in Figure 3c is the resulting synthesized HR image



**Fig. 4.** (a,d) Input LR image. (b,e) Original HR image. (c,f) The synthesized HR image. (d-f) are the image regions in actual size from (a-c), respectively.

after running our algorithm. In order to visually compare the results, Figure 31 shows a zoomed region of each of the images in shown above in the same order.

A second example is depicted in Figure 4. As the previous example, the first row shows from left to right, the original HR image, the input LR image and the synthesized HR image after running our algorithm. In the second row, zoomed versions of the same region corresponding to the above images are shown.

## 5 Conclusions

We have presented an innovative approach for the problem of super-resolution for the limited case, where only a single frame (picture) is given as an input. The method exploits the fact that the inter-relationships between the high frequencies in the small-sized images are similar to the inter-relationships between the high frequencies in bigger images. It is for this reason that our method is hierarchical simultaneously by the scales that use inter-scale characteristics and describe these characteristics not only by themselves, but in terms of its inter-relationships with other characteristics. The experimental results shown here demonstrate the feasibility of the method.

## References

1. Baker, S., Kanade, T.: Limits on super-resolution and how to break them. *IEEE Transactions on PAMI* 24(9), 1167–1183 (2002)
2. Candocia, F.M., Principe, J.C.: Super-resolution of images based on local correlations. *IEEE Trans. on Neural Networks* 10(2), 372–380 (1999)
3. Mahesh, B.: Chappalli. Image enhancement using SGW superresolution and iterative blind deconvolution. PhD thesis, The Pennsylvania State University (2005)
4. Chiang, M.C., Boulton, T.E.: Efficient super-resolution via image warping. *Image and Vision Computing* 18, 761–771 (2000)
5. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. *Computer Graphics and Applications, IEEE* 22(2), 56–65 (2002)
6. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. *International Journal of Computer Vision* 20(1), 25–47 (2000)
7. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on PAMI* 6, 721–741 (1984)
8. Hardie, R.C., Barnard, K.J., Armstrong, E.E.: Joint map registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Trans. on Image Processing* 6(12), 1621–1633 (1997)
9. Jiji, C.V., Chaudhuri, S.: Single-frame image super-resolution through contourlet learning. *EURASIP Journal on Applied Signal Processing* (2006)
10. Kim, S.P., Bose, N.K., Valenzuela, H.M.: Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Trans. on Acoustics, Speech and Signal Processing* 18(6), 1013–1027 (1990)
11. Ni, K.S., Kumar, S., Vasconcelos, N., Nguyen, T.Q.: Single image superresolution based on support vector regression. In: *IEEE International Conference on Acoustics, Speech and Signal Processing* (2006)

12. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE* 20(3), 21–36 (2003)
13. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc, San Francisco (1988)
14. Su, K., Tian, Q., Xue, Q., Sebe, N., Ma, J.: Neighborhood issue in single-frame image super-resolution. In: *IEEE International Conference on Multimedia and Expo, ICME* (July 2005)
15. Tsai, R.Y., Huang, T.S.: Multiframe image restoration and registration. *Advances in Computer Vision and Image Processing*, 317–339 (1984)
16. Weiss, Y.: *Belief propagation and revision in networks with loops*. Technical report, Berkeley Computer Science Dept. (1998)

# Shadows Attenuation for Robust Object Recognition

J. Gabriel Aviña-Cervantes<sup>1</sup>, Leonardo Martínez-Jiménez<sup>1</sup>, Michel Devy<sup>2</sup>,  
Andres Hernández-Gutierrez<sup>1</sup>, Dora L. Almanza<sup>1</sup>, and Mario A. Ibarra<sup>1</sup>

<sup>1</sup> University of Guanajuato.

Facultad de Ingeniería Mecánica, Eléctrica y Electrónica,  
Avenida Tampico 912. Salamanca, Guanajuato. 36730, México  
{avina,luzdora,ibarram}@salamanca.ugto.mx

<http://www.ugto.mx>

<sup>2</sup> Laboratoire d'Analyse et d'Architecture des Systèmes,

LAAS-CNRS, 7, Avenue du Colonel Roche,  
31077 Toulouse Cedex 4, France

{gavina, michel}@laas.fr

<http://www.laas.fr>

**Abstract.** Shadows are useful for synthetic images in order to increase extrinsically reality in image generation. However, in natural images, object recognition and segmentation are often negatively affected by cast shadows. Since shadows are a physical phenomena observed in most natural scenes, we propose a fast and reliable procedure to detect and attenuate shadows effects based on color/brightness density. Detected shadows are attenuated by modifying locally brightness and color that have the same color/brightness density. Some color artifacts (false colors on shadows) produced by the acquisition devices have been detected and discussed, and it has been noticed that they may affect some of the classical shadow removal methods. Finally, some experimental results of the proposed shadow attenuation method in real images are presented and evaluated.

## 1 Introduction

Shadows are one of the principal factors affecting machine vision performance in outdoor scenes. Many applications such as content based image retrieval, based on the knowledge of color or texture features, are negatively affected by strong cast shadows. For instance, in applications using color segmentation, object and shadows are often associated to the same object, generating serious problems to the capabilities of the recognition phase [1]. Furthermore, most image recognition procedures are based on texture as their principal feature. Texture has been defined as a local arrangement of irradiance patterns projected over a surface mosaic of perceptually homogeneous radiances (high concentration of localized spatial frequencies). The analysis of texture in natural environments turns out to be very difficult, being shadows one of the problems to overcome.

Shadows are produced when there are objects between a light source and a surface, this produces an effect of division on the surface, as if it had two surfaces.



Due to the reduction of the illumination level on the affected region, regions having high contrast are sometimes associated to the object. Consequently, these illumination differences over the same surface produce serious problems in image processing, especially in natural images; object recognition and segmentation are often negatively affected by cast shadows [2,3].

Due to the effects of shadows in image processing tasks, there is some researches focused on shadow removal methods. The majority of shadow removal methods have a thresholding step in order to find a shadow map of the edges. Other methods work controlling the properties of gamma, contrast and brightness of the image, to reduce the effect of the low illumination [4,5]. Figure 1 shows some classical applications: visual robot navigation and image interpreting where shadows produce crucial failure in the routines [6]. In order to reduce, the negative effects of low illumination levels, other methods work controlling some other properties on the image, such as nonlinear gamma, contrast and brightness adjustments [2].



(a) Scene Interpretation

(b) Robot Visual Navigation

**Fig. 1.** Cast shadows interference in image understanding and object recognition

When an image has regions affected by shadows, we may assume that the shadow affects only the brightness on the projected surface significantly and the color properties are only alter lightly [7]. Moreover, some color spaces are more suitable for shadows removals than others, *e.g.*, color segmentation. There are color spaces with uncorrelated components ( $I_1, I_2, I_3$ ), some other use non-linear transformations ( $HSI, CIE-LUV, CIE-LAB$ ) and in our application probably the most useful spaces are those which allow separate luminance from chrominance components [8,9].

The CIE Lab color space is able to separate the color information from the brightness. The  $L$  channel is the luminance channel, the  $a$  channel is a scale going from the green color with negative values to the red color with positive values, and the  $b$  channel is a scale that goes from the blue color with negative values to yellow color with positive values [10,11]. CIE Lab color space has been adopted for an important part of our experiments.

This paper is organized as follows, in section 2 the basis of our shadows attenuator is presented, in section 3 experimental results are shown and discussed and finally, in section 4 general conclusions of this work are presented.

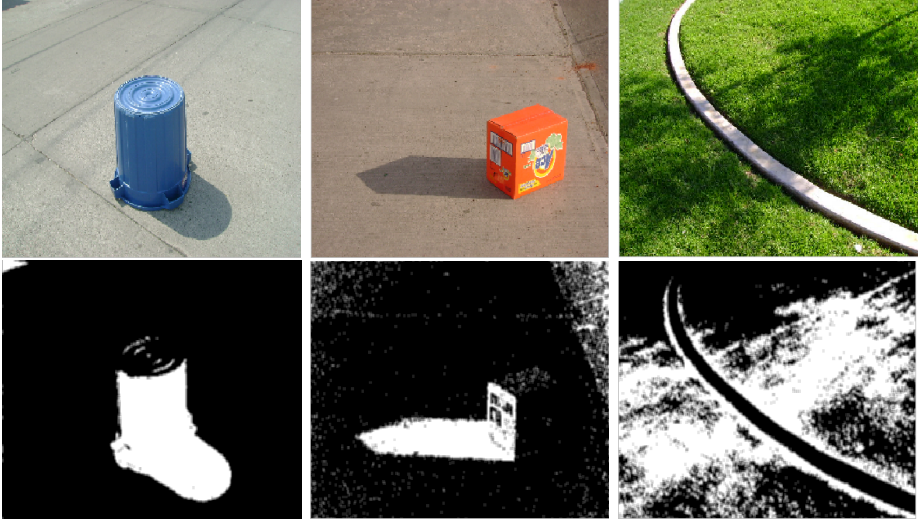


Fig. 2. Inverse luminances for Shadowed Images

## 2 Shadow Attenuator

Considering the assumption that small changes on color channels  $a$  and  $b$  are produced by shadows effects, we can compare two regions, in order to look for regions with similar color <sup>1</sup> density [12][5]. If we have regions  $s_1$  and  $s_2$ , we can say that  $s_1$  is a shadow region similar in color to  $s_2$  if both of them have similar color densities and  $s_1$  has smaller brightness level than  $s_2$ . Consequently, we must compare the region  $s_1$  with a limited number of different  $s_2$  random regions around  $s_1$ , the comparison process stops for region  $s_1$  when a similar color density is found in region  $s_2$ , and when the luminance  $s_1$  is less than the luminance of  $\alpha s_2$ . Where  $\alpha$  is a threshold number in the range  $[0, 1]$  that let us discriminate shadow differences based on luminance and chrominance statistics on the image. In other words, a low intensity region suspected to be a shadow is compared chromatically against a set of regions located in its neighborhood giving a certain distance and a seek criterion.

The saturation of the image in the CIE Lab color space is given by the following equation,

$$c = \sqrt{a^2 + b^2} \quad (1)$$

where  $a$  and  $b$  are the chromatic components in the CIE-LAB color space. For neighboring regions,  $s_1$  and  $s_2$  chromatic features  $c_1$  and  $c_2$  have been computed over all both regions, by using equation 1.



It has been considered that chromatics  $c_1$  and  $c_2$  represent similar color densities, if both  $c_1$  and  $c_2$  over delimited regions have closed values. In this way,  $c_1$  and  $c_2$  are supposed to have similar numerical values if:

$$v_{cmin} \leq \frac{c_1}{c_2} \leq v_{cmax} \tag{2}$$

where  $v_{cmin}$  and  $v_{cmax}$  represent similarity boundaries estimated statistically over the image or fixed by the user.



**Fig. 3.** Corrected Luminances for Shadowed Images

In particular for the brightness channel in the region  $s_1$ ,  $L_{s_1}$  could be considered as belonging to a shadowed region if  $L_{s_1}$  is smaller than the luminance in the region  $s_2$  affected by a parameter,  $\alpha L_{s_2}$ . However, experimental observations are shown that a shadow is often a region with a lower illumination than the overall averaged illumination on the scene or around its neighborhood (regions delimiting the suspected area). This can be expressed by the next relation between a luminances ratio,

$$\frac{L_{s_1}}{L_{s_2}} < \alpha \tag{3}$$

where  $\alpha$  is estimated dynamically using the mean of the luminance on the scene, *i.e.*,  $E\{L\}$ . Given two regions  $s_1$  and  $s_2$  in a image, we can say  $s_1$  is a region affected by a shadow with similar color density to  $S_2$  if the next three conditions are true:

$$L_{s_1} < E\{L\}$$

$$v_{min} \leq \frac{c_1}{c_2} \leq v_{max} \quad (4)$$

$$\frac{L_{s_1}}{L_{s_2}} < \alpha$$

where  $0 < \alpha < 1$ . The proposed algorithm takes a region  $s_1$  which is considered as a shadow region, around it, a second region  $s_2$  is searched, both regions  $s_1$  and  $s_2$  must satisfy the three conditions given previously in order to start the shadow attenuation procedure. If the first condition is not satisfied,  $s_1$  is not conceived as a shadow, because this region has a high illumination level (a shadow region is assumed as a low illumination region) and consequently  $s_1$  is discarded as a shadow region.

If the first condition is true, but any other of the two remainder conditions are not satisfied, the algorithm seeks for another suitable region  $s_2$  over random coordinates  $(m, n)$  around the suspected region. At the moment that a shadow region in the scene is correctly identified, it is necessary to modify automatically the brightness level in areas affected (*e.g.*, increasing luminance). In order to find the new brightness value for every shadow, a ratio for each couple  $L_{s_1}$ ,  $L_{s_2}$  may be found.

$$ratio = \frac{L_{s_1}}{L_{s_2}} \quad (5)$$

Adjusting Luminance is achieved by changing  $L_{s_2}$  with a new value  $L_{1new}$ , a formula for  $L_{1new}$  can be expressed as follows,

$$L_{1new} = \frac{L_{s_1}}{ratio} \quad (6)$$

In our experiments, we compute the average of all ratios obtained in the image. These ratios are substituted by the average  $E\{ratio\}$  and multiplied by a factor of adjustment. Equation 5 is modified, and in this way, equation 7 is generated for a set of regions composing the neighborhood of the detected region.

$$L_{1new} = \frac{1}{\beta E\{ratio\}} L_{s_1} \quad (7)$$

If suspected region  $s_1$  is not surely identified as a cast shadow, its luminance channel  $L_{s_1}$  should not be changed or affected. Thus, shadows coverage (dark surface) is considered uniformly distributed or illuminated is a strong assumption which is valid in many circumstances but a non uniform shadows removal may be necessary in some critical applications. However, for robotics applications

and some other machine recognition tasks where real-time routines are needed, the proposed approach is reliable and useful [13]. In the next section, some experimental results will be presented and commented.

### 3 Experimental Results

In order to evaluate the capabilities of our algorithm, experimental results have been classified in three kind of tests. In the first category, the capacity to detect shadows in the images is correctly identified, in the second test the luminance in the output image over detected shadows is balanced and finally the capacity to attenuate shadows in RGB color image is shown.



Fig. 4. Corrected Images by proposed method

#### 3.1 Shadow Detection Phase

With a light modification in the algorithm, the pixels classified as shadow can be identified, painting detected pixels as shadows with maximum value (white) and zero for the rest elements on the image (black), see Fig. 2.

The results show that the proposed method is efficient to detect shadows in the majority of the images; it detected 28 of a total of 31 test images. This research shows that it is necessary only a maximum of 15 comparisons to find the appropriate region  $s_2$  needed to balance shadow brightness or considerate  $s_1$  as a non shadow region.

In the images where the method failed, this situation was principally due to a reduced value in the contrast between shadows and the scene, *i.e.*, shadow is not so strong and the procedure could not discriminate it from the rest of the surface or when the shadow area covers the majority of the image surface (more than the

half of the total area). This non detection is easily explained due to the nature of the proposed algorithm, since non shaded area is above the averaged on the illumination. Another particular case occurs when the algorithm does not find a similar color density area with respect to the suspected region and consequently, the original illumination can not be adjusted and balanced.

### 3.2 Adjusting Luminance Image

Experimental results concerning the modification of the luminance image show that, with the correct value of the parameter  $\beta$ , the effect of the shadow in the luminance image is efficiently reduced or attenuated. In figure 4 (a), (b) and (c) images are the original images, next images (d), (e) and (f) are luminance  $L$  original images and (g), (h) and (i) are the luminance adjusted or output images. Some values for the parameter  $\beta$  are estimated for this sample, image (g) was obtained using  $\beta = 1$ , image (h) with  $\beta = 0.93$  and image (i) with  $\beta = 0.80$ .

In images where the shadows are very strong, it is only possible to reduce the shadow effects with very small values of  $\beta$  (0.5 approximatively). In some images, shadow has not the same value along certain perspectives of the image, in this case the method does not work appropriately, because the algorithm finds the average value product in the reduction of illumination due to the effect of shadowing to correct affected areas, *i.e.*, a correction is uniformly applied under a region non uniformly shadowed.

A negative effect produced by modifying the luminance image may consist in the presence of white-borders, which are caused by the appearance of different levels of illumination due to changes in the angle of light source (penumbra), in images where the shadow is distant or small, penumbra does not affect importantly shadows illumination. However, in images where the shadow is close or big, the penumbra area is important, and the differences within the shadow are evident. This penumbra phenomena may be reduced by a smoothing filter focused on the homogenization of dark zones.

Once the effect of the shadows in the luminance image are reduced, the color image can be reconstructed, with the luminance image adjusted and balanced, nevertheless, before the re-compose the three color channels, it is needed to fix the luminance image in order to have values in the range  $[0, 0.95]$  to reduce over-brightness values in the image, this reduction helps to avoid overflow on some values in the output image and increase the quality of the resulting image. Finally, it is necessary to use an average filter in order to avoid the possible false color that could be generated.

Experimental results have been classified for all process inside of four categories:

1. Shadow was detected and strongly attenuated.
2. Shadow was detected and attenuated.
3. Shadow was detected but not attenuated significantly.
4. Shadow was not detected.



(a) Original Image

(b) Shadow attenuated images

**Fig. 5.** Shadows influenced by neighbor objects producing false color

The results of the 31 completely different images that were processed give the performance table [11](#), where it is possible to notice that the majority of the images were at least detected satisfactory and attenuated. And, almost a third part of them were strongly attenuated. From the total of 31 images, only 4 show problems produced by acquisition devices, where false colors are produced mainly by the white point color balancing algorithms built in the cameras and which very often produce false color on the surfaces affected by shadows (see Fig. [5](#)). In some cases the shadow is so strong, so it could destroy the texture and color, thus it is impossible to restore the color information under the shade. Another problem consists in the influence of external elements located around the object which produces non-uniform shadows (including also the same object), because of these objects contribute with the spectral distribution of the lighting near of the object. Therefore, complex shadow has color components conformed by several light reflections coming from many sources. Moreover, performance of our algorithm was qualitatively compared with respect to the most popular shadow removal method, FHD [10](#). Principal difference consists in preserving texture quality with the proposed approach while FHD method corrects globally shadows with no complex artifacts but affecting negatively textured regions. This texture degeneration is produced by the morphologic algorithm to enhance the shadow edges, see Fig. [6](#).

**Table 1.** Performance of the algorithm

Category	Number of images
Shadows detected and strongly attenuated	10
Shadows detected and attenuated	11
Shadows detected	6
Shadows no detected	4





**Fig. 6.** Qualitative comparison with the FHD algorithm and proposed algorithm. Vertically, (a) original images, (b) FHD method, and (c) proposed approach.

## 4 Conclusions

In this paper, a shadow detector and attenuator algorithm has been proposed. Experimental results show that it is possible to identify and efficiently correct in an efficient way shadows in the 80% of our test images. Our procedure fails primordially when color covered by the shadow regions is not similar to the surfaces around the shadows. It is important to express the importance of the sensor (cameras) properties that can modify the chromatic properties over the shadows as well as the reflectivities produced by the interaction of the light and the objects around of the shadows and penumbra regions. Therefore, the proposed approach is a reliable and fast procedure which can be used as a pre-processing stage on pattern recognition procedures affected by shadows, which occur mainly in outdoor applications as robot visual navigation in natural environments.

**Acknowledgement.** We would like to thank to CONCYTEG and PROMEP for the funding obtained through the projects: "Application of the chromatic adaptation models into the texture analysis of natural scenes" (code 07-16-K662-061/03, CI3038307) and "Visual navigation of robots in agricultural environments based on autonomous road detection and tracking" (PTC 103.5/05/1924).

## References

1. Rosin, P.L., Ellis, T.: Image difference threshold strategies and shadow detection. In: Sixth British Machine Vision Conference, BMVC, Birmingham, Uk, pp. 347–356 (1995)
2. Hsieh, J.-W., Hu, W.-F., Chang, C.-J., Chen, Y.-S.: Shadow elimination for effective moving object detection by gaussian shadow modeling. *Image and Vision Computing* 21(6), 505–516 (2003)
3. Liu, Z., Huang, K., Tan, T., Wang, L.: Cast Shadow Removal with GMM for Surface Reflectance Component. In: Proceedings of the 18th International Conference on Pattern Recognition, pp. 727–730 (2006)
4. Stauder, J., Mech, R., Ostermann, J.: Detection of moving cast shadows for object segmentation. *IEEE Transactions on Multimedia* 1(1), 65–76 (1999)
5. Baba, M., Mukunoki, M., Asada, N.: Shadow Removal from a Real Image Based on Shadow Density. In: 31st International Conference on Computer Graphics and Interactive Techniques (2004)
6. Gu, X., Yu, D., Zhang, L.: Image Shadow Removal Using Pulse Coupled Neural Network. *IEEE Transaction on Neural Networks* 16(3), 692–698 (2005)
7. Tadjine, H.H., Joubert, G., Mouattamid, S.M.: Colour object detection with shadow using virtual electric field. In: International conference on image and signal processing, ICISP, Agadir, Morocco, vol. 1, pp. 53–60 (2003)
8. Finlayson, G., Hordley, S., Drew, M.: Removing Shadows from Images Using Retinex. In: 10th Color Imaging Conference (2002)
9. Buluswar, S.D., Draper, B.A.: Color Recognition in Outdoor Images. In: Proceedings of the Sixth International Conference on Computer Vision, pp. 171–177 (1998)
10. Finlayson, G.D., Hordley, S.D., Drew, M.S.: Removing Shadows from Images. In: Tistarelli, M., Bigun, J., Jain, A.K. (eds.) ECCV 2002. LNCS, vol. 2359, pp. 823–836. Springer, Heidelberg (2002)
11. Jiang, C., Ward, M.O.: Shadow Segmentation and Classification in a Constrained Environment. *CVGIP: Image Understanding* (1994)
12. Salvador, E., Cavallaro, A., Ebrahimi, T.: Shadow Identification and Classification using Invariant Color Models. *International Conference on Acoustics, Speech, and Signal Processing* 3, 1545–1548 (2001)
13. Mateus, D., Avina-Cervantes, J.G., Devy, M.: Robot visual navigation in semi-structured outdoor environments. In: IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain (2005)

# Fuzzy Directional Adaptive Recursive Temporal Filter for Denoising of Video Sequences

Alberto Rosales-Silva<sup>1</sup>, Volodymyr Ponomaryov<sup>1</sup>, and Francisco Gallegos-Funes<sup>2</sup>

National Polytechnic Institute of Mexico  
Mechanical and Electrical Engineering Higher School

<sup>1</sup> ESIME-Culhuacan; Av. Santa Ana 1000, Col. San Francisco Culhuacan,  
04430, Mexico D.F., Mexico  
vponomar@ipn.mx

<sup>2</sup> ESIME-Zacatenco; Av. IPN s/n, U.P.A.L.M. Col. Lindavista,  
07738, Mexico D.F., Mexico  
fgallegosf@ipn.mx

**Abstract.** In this paper we present the fuzzy directional adaptive recursive temporal filter for denoising of video sequences. The use of spatial-temporal information is considered more efficient in presence of fast motion and noise. We connect the differences between images, such as, the angle deviations to obtain several parameters to apply them in the proposed algorithm to detect and differentiate movement in background of noise. Extensive simulation results have demonstrated that the proposed fuzzy filter can consistently outperforms other filters by balancing the tradeoff between noise suppression and detail preservation.

## 1 Introduction

The motion detection problem is very complex due it is not always easy to distinguish illumination changes from real motion and because of the aperture problem. However, in several applications it is sufficient to detect changes in the scene rather than actual motion, and even to detect only some of the changes [1].

In this paper, we propose a method to distinguish image noise and changes due to motion of camera zoom and movement in the scene present in video sequences. We have developed the mathematical operations to consume less time, it can be achieved dividing different operations depending of parameters obtained using fuzzy logic membership functions. This permits to realize robust noise suppression and movement detection.

The goal of the proposed method is to use adaptive threshold that is adapted to the local pixel statistics and the spatial pixel context. The proposed method is insensitive to noise; it is locally adaptive to spatially varying noise levels. The presented method uses data incoming during long period of time, and the threshold is adapted to both temporal and spatial information [1-3]. The noise is not labelled as motion, it is not so important if not every single changed pixel of an object is detected. However, in the case of motion detection during the denoising processing, where the detection result is used for temporal filtering, undetected changes in an object can lead to motion blur,



but in the same time if some noise is labelled as motion it is no so critical. Using fuzzy logic techniques we aim at defining a confidence measure with respect to the existence of motion, to be called hereafter “motion confidence” [2].

In section 2 we present the proposed algorithm for simultaneous motion detection and video denoising. Experimental results are given in section 3. Finally we draw our conclusions in section 4.

## 2 Proposed Algorithm

In this section we present our proposed algorithm for simultaneous video denoising and motion detection. The proposed algorithm is depicted in Figure 1.

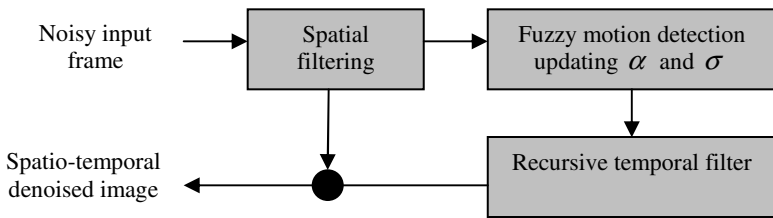


Fig. 1. Block diagram of proposed algorithm

### 2.1 Spatial Filtering

The Gaussian estimation algorithm is used to suppress the noise in the first stage, *Step 1)* IF  $\theta_c \leq F/255$  THEN Histogram is increased in “1”, else is “0”.

*Step 2)* Compute probabilities for each one of these samples:

$$p_i = \text{Histogram}_i / S, i=0, \dots, 255. \tag{1}$$

*Step 3)* Obtain standard deviation  $\sigma'_T$  :

$$\mu = \sum_{i=0}^{255} i \cdot p_i; \sigma^2 = \sum_{i=0}^{255} (i - \mu)^2 \cdot (p_i), \sigma'_T = \sqrt{\sigma^2} \tag{2}$$

where  $S = \sum_i \text{Histogram}_i$ ,  $\theta_c = A(\bar{y}, y_c)$  is the angle deviation,  $\bar{y} = 1/N \sum_{i=1}^N y_i$  is the mean value with  $i=1, \dots, N, N=9$ , and  $y_c$  is the central pixel.

### 2.2 Fuzzy Motion Detection

Each plane of video sequence is processed in an independent way, and the parameters  $\sigma'_T = \sigma'_{red} = \sigma'_{green} = \sigma'_{blue}$  are adapted along the video sequence. The angle deviations  $\theta_i = A(x_i, x_c)$  are calculated [4], where  $i=1, \dots, N-1, N=9$ , and  $i \neq \text{central pixel}$ .

Then, we detect uniform regions using the mean weighted filtering algorithm [5]:

Step 1) IF  $\theta_2$  AND  $\theta_4$  AND  $\theta_7$  AND  $\theta_5 \geq \tau_1$  THEN,

$$y_{out} = \sum_{\substack{i=1 \\ i \neq c}}^{N-1} x_i \left[ \frac{2}{(1 + e^{\theta_i})^r} \right] + x_c \left/ \sum_{i=1}^{N-1} x_i \left[ \frac{2}{(1 + e^{\theta_i})^r} \right] \right. + 1 \tag{3}$$

Step 2) IF  $\theta_6$  AND  $\theta_3$  AND  $\theta_1$  AND  $\theta_8 \geq \tau_1$  THEN, Use eq. (3).

This algorithm smoothes Gaussian noise very fast. The central pixel has the highest weight with “1” to preserve some features in uniform regions. A 5x5 window is used to estimate the standard deviation in same way as Gaussian estimation algorithm to obtain the values  $\sigma_T$ , and then we compare with the values  $\sigma'_T$  to have a similarity value for each sample and to have a criterion in perform more or less filtering charge. We obtained that if  $\sigma'_T < \sigma_T$ , then  $\sigma_T = \sigma'_T$ , otherwise  $\sigma'_T = \sigma_T$ , where  $T$  can be the component red, green, or blue, that permits to improve temporal filtering algorithm. By optimum PSNR and MAE we defined a threshold  $Th_T = 2\sigma_T$  to preserve some features in a spatially filtered frame that will be used in temporal algorithm.

### 2.3 Fuzzy Vector Gradient Values

For each pixel  $(i,j)$  of the any component image, we use a 3x3 neighbourhood window as illustrates Figure 2.

If  $A_T$  denotes one component input image, the gradient can be defined as,

$$\nabla_{(k,l)} A_T(i, j) = |A_T(i + k, j + l) - A_T(i, j)| \text{ with } k, l \in \{-1, 0, 1\} \tag{4}$$

where the pair  $(k,l)$  corresponds to one of the eight directions that are called *the basic gradient values* [2], and  $(i,j)$  is called the centre of the gradient.

To avoid blurred in presence of an edge, we use one basic gradient for each direction and two related gradient values. The three gradient values for a certain direction are finally connected together into one single value called fuzzy gradient value.

We take pixels as vectors to have directional process, taking the same procedure as in gradient values. By this way we obtain Fuzzy vector gradient values that are

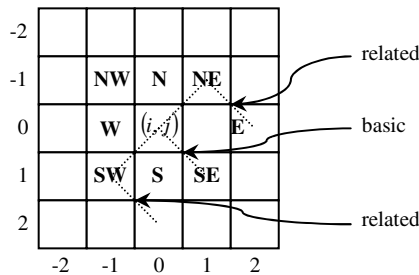


Fig. 2. Basic and related gradients and vector gradients

defined by *Fuzzy Rule 1*. The two related gradient values in the same direction and the basic gradients are determined by the centres making a right-angle with the direction of the corresponding basic gradient [3].

To determine the basic vector gradient values we use the following algorithm,  
*Step 1)* IF  $\nabla_{\gamma\beta} < T_{s\beta}$  THEN compute angle deviation in the direction  $\alpha_{\gamma\beta}$  and obtain weight value,

$$\alpha'_{\gamma\beta} = 2 / \left( 1 + e^{\alpha_{\gamma\beta}} \right)^r \tag{5}$$

*Step 2)* Obtain basic vector gradient using membership function.

*Step 3)* IF  $\nabla_{\gamma\beta} > T_{s\beta}$  THEN  $\mu_{BIG} = 0$ .

where  $T_{s\beta} = Th_r$ ,  $\gamma = NW, N, NE, E, SE, S, SW, W$  [6],  $\beta = red, gree, blue$  and  $r = 1$  channels in video sequence, the membership function is  $\mu_{BIG} = \max(x, y)$  with  $x = \alpha'_{\gamma\beta}$  and  $y = (1 - \nabla_{\gamma\beta} / T_{s\beta})$ . To obtain the angle deviation in each plane of the image we select to work in the angle formed by vectors in only one coordinate [4]:

$$\alpha = \cos^{-1} \left( (r_1 r_2 + g_1 g_2 + b_1 b_2) / \sqrt{(r_1^2 + g_1^2 + b_1^2)(r_2^2 + g_2^2 + b_2^2)} \right) \tag{6}$$

where,  $(r_1, g_1, b_1)$  and  $(r_2, g_2, b_2)$  are coordinates of two pixels.

To determine related vector gradients the procedure is the following:

*Step 1)* IF  $\nabla_{\gamma\beta(R1,R2)} < T_{s\beta}$  THEN compute angle deviation in direction  $\alpha_{\gamma\beta(R1,R2)}$  and obtain weight value [5],

$$\alpha'_{\gamma\beta(R1,R2)} = 2 / \left( 1 + e^{\alpha_{\gamma\beta(R1,R2)}} \right)^r \tag{7}$$

*Step 2)* Obtain related vector gradient using membership function.

*Step 3)* IF  $\nabla_{\gamma\beta(R1,R2)} > T_{s\beta}$  THEN  $\mu_{BIG} = 0$ .

where  $(R1, R2)$  are the related vector gradients and the membership function is  $\mu_{BIG} = \max(x, y)$  with  $x = \alpha'_{\gamma\beta(R1,R2)}$  and  $y = (1 - \nabla_{\gamma\beta(R1,R2)} / T_{s\beta})$ .

The fuzzy rule 1 is defined as,

Fuzzy Rule 1: defining the fuzzy vector gradient value $\nabla_{\gamma\beta}^F A_{\beta}(i, j)$ ,
IF $\nabla_{\gamma\beta}$ is BIG AND $\nabla_{\gamma\beta R1}$ is BIG, OR $\nabla_{\gamma\beta}$ is BIG AND $\nabla_{\gamma\beta R2}$ is BIG, THEN $\nabla_{\gamma\beta}^F A_{\beta}(i, j)$ is BIG,

where  $\nabla_{\gamma\beta}$  is the basic vector gradient value, and  $\nabla_{\gamma\beta R1}$  and  $\nabla_{\gamma\beta R2}$  are two related vector gradient values for the direction  $\gamma$  in the channel  $\beta$ .

If basic and related vector gradients are close enough, in absolute difference or norm, or in a vector criterion in angle distances (we change gradient values to vector gradient values), this proposal is developed to obtain robust parameters to understand better the nature of pixels in a window processing. Under this criterion we will have values denoted as fuzzy vector gradients that means nearby in pixels related, and they

are helpful to suppress Gaussian noise corruption. So, suppression is done by a weighted mean procedure where nearby close to 1 have bigger weights in the algorithm due to proposed procedure used in membership function. This suppresses noise more efficiently but smoothes details and edges, in our complete algorithm the temporal filtering are designed. The reference values where found modifying their parameters according to optimum PSNR and MAE values. Spatial algorithm presents good results in noise suppression compared with algorithms found in literature [1-3].

The weighted mean algorithm is implemented by:

$$y_{out} = \sum_{\substack{i=0 \\ i \neq c}}^{N-1} y_{\gamma} \cdot x_{\mu} \bigg/ \sum_{i=0}^{N-1} x_{\mu} \tag{8}$$

where mean value is found doing multiplication of fuzzy vector gradient value with his respective pixel in that direction  $\gamma$ .

### 2.4 Temporal Filtering

The proposed fuzzy logic recursive motion detector with temporal algorithm is explained in this section. The reference values of spatial filter presented above are used in the final stage in the proposed filter. Only the past and present frames are used to avoid dramatic charge in memory requirements and time processing. The fuzzy logic rules are used in each plane of two frames in independent way.

We found angle deviations and gradient values by the central pixel in present frame respect to his neighbours in past frame in each plane of the frames,

$$\theta_i^1 = A(x_i^A, x_c^B) ; \nabla_i^1 = |x_i^A - x_c^B| ; i = 1, \dots, N ; N = 9 \tag{9}$$

where  $x_c$  is central pixel in present frame, and  $A$  and  $B$  are past and present frames by planes, respectively.

We define the membership functions to obtain a value that indicates the degree, in which a certain gradient value or vector value matches the predicate. If a gradient or a vector value have membership degree one, for the fuzzy set SMALL, it means that it is SMALL for sure in this fuzzy set. Selection of this kind of membership functions is follow from nature of pixels, where a movement is not a linear response, and a pixel has different meanings in each frame of video sequence.

Membership functions SMALL and BIG for angles and gradients are given by [7]:

$$\mu_{SMALL}(M) = \begin{cases} 1 & M < med \\ \exp\left(-\frac{(M - med)^2}{2\sigma_s^2}\right) & \text{otherwise} \end{cases} \tag{10}$$

$$\mu_{BIG}(M) = \begin{cases} 1 & M < med \\ \exp\left(-\frac{(M - med)^2}{2\sigma_B^2}\right) & \text{otherwise} \end{cases} \tag{11}$$

where  $\sigma_s^2 = 0.1$ ,  $\sigma_b^2 = 1000$ , and  $M$  can be the angle  $\theta$  or gradient  $\nabla$ , for angles  $med=0.2$  and  $med=0.9$  for membership functions SMALL and BIG, respectively, and for gradients  $med=60$  and  $med=140$  for SMALL and BIG functions, respectively.

Now we use Fuzzy Rules 2, 3, 4, and 5 to acquire corresponding values:

<b>Fuzzy Rules</b>
<p><b>Fuzzy Rule 2:</b> Defining the fuzzy gradient-vector value <math>SBB(x, y, t)</math>.</p> <p>IF <math>\theta^1(x, y, t)</math> is SMALL AND <math>\theta^2(x, y, t)</math> is BIG AND <math>\theta^3(x, y, t)</math> is BIG AND <math>\nabla^1(x, y, t)</math> is SMALL AND <math>\nabla^2(x, y, t)</math> is BIG AND <math>\nabla^3(x, y, t)</math> is BIG THEN <math>SBB(x, y, t)</math> is true.</p>
<p><b>Fuzzy Rule 3:</b> Defining the fuzzy gradient-vector value <math>SSS(x, y, t)</math>.</p> <p>IF <math>\theta^1(x, y, t)</math> is SMALL AND <math>\theta^2(x, y, t)</math> is SMALL AND <math>\theta^3(x, y, t)</math> is SMALL AND <math>\nabla^1(x, y, t)</math> is SMALL AND <math>\nabla^2(x, y, t)</math> is SMALL AND <math>\nabla^3(x, y, t)</math> is SMALL THEN <math>SSS(x, y, t)</math> is true.</p>
<p><b>Fuzzy Rule 4:</b> Defining the fuzzy gradient-vector value <math>BBB(x, y, t)</math>.</p> <p>IF <math>\theta^1(x, y, t)</math> is BIG AND <math>\theta^2(x, y, t)</math> is BIG AND <math>\theta^3(x, y, t)</math> is BIG AND <math>\nabla^1(x, y, t)</math> is BIG AND <math>\nabla^2(x, y, t)</math> is BIG AND <math>\nabla^3(x, y, t)</math> is BIG THEN <math>BBB(x, y, t)</math> is true.</p>
<p><b>Fuzzy Rule 5:</b> Defining the fuzzy gradient-vector value <math>BBS(x, y, t)</math>.</p> <p>IF <math>\theta^1(x, y, t)</math> is BIG AND <math>\theta^2(x, y, t)</math> is BIG AND <math>\theta^3(x, y, t)</math> is SMALL AND <math>\nabla^1(x, y, t)</math> is BIG AND <math>\nabla^2(x, y, t)</math> is BIG AND <math>\nabla^3(x, y, t)</math> is SMALL THEN <math>BBS(x, y, t)</math> is true.</p>
<p>where <math>\theta^r(x, y, t)</math> are angles values, <math>\nabla^r(x, y, t)</math> are gradient values, and <math>r = 1, 2, 3</math>.</p>

From the result of each fuzzy rule (2–5) we compare these values as follows:

<b>Algorithm to Fuzzy Rule <math>SBB(x, y, t)</math></b>
<p>If <math>SBB(x, y, t)</math> is the biggest value found from the others:</p> <p>Step 1) IF <math>\{(SBB(x, y, t) &gt; SSS(x, y, t)) \text{ AND } (SBB(x, y, t) &gt; BBB(x, y, t)) \text{ AND } (SBB(x, y, t) &gt; BBS(x, y, t))\}</math> THEN Weighted mean using <math>SBB(x, y, t)</math>,</p> $y_{out} = \frac{\sum p^A(x, y, t) \cdot SBB(x, y, t)}{\sum SBB(x, y, t)}$ <p>Step 2) Update standard deviation for next frames to divide details from uniform regions.</p>

where  $SBB(x, y, t)$  value says that central pixel is in movement because of big differences in local and gradient values,  $p^A(x, y, t)$  is each pixel in last frame that fulfil with the IF condition, and  $y_{out}$  is the output filtered in spatial and temporal filtering.

To update standard deviation we need different values by each condition in our algorithm to characterize each region of the image. This is achieved using the expression above, and it is always used after by each Fuzzy Rule to update the parameter:

$$\sigma_T' = (\alpha \cdot \sigma_{TOTAL}) + (1 - \alpha) \cdot (\sigma_T') \tag{12}$$

where  $T = red, green, blue$ ,  $\alpha = \alpha_{SBB} = 0.875$ , and  $\sigma_{TOTAL} = (\sigma_{red} + \sigma_{green} + \sigma_{blue}) / 3$ .

<p><b>Fuzzy Rule <math>SSS(x,y,t)</math></b></p> <p>If <math>SSS(x,y,t)</math> is the biggest value found from the others:</p> <p><i>Step 1</i>) IF <math>\{(SSS(x,y,t) &gt; SBB(x,y,t)) \text{ AND } (SSS(x,y,t) &gt; BBB(x,y,t)) \text{ AND } (SSS(x,y,t) &gt; BBS(x,y,t))\}</math> THEN Weighted mean using <math>SSS(x,y,t)</math>,</p> $y_{out} = \frac{\sum (p^A(x,y,t) \cdot 0.5 + p^B(x,y,t) \cdot 0.5) \cdot SSS(x,y,t)}{\sum SSS(x,y,t)}$ <p><i>Step 2</i>) Update standard deviation for next frames to divide details from uniform regions.</p>
---

where  $SSS(x,y,t)$  shows that a central pixel is not in movement because of small differences in all directions, that is why we use pixels in both frames,  $p^A(x,y,t)$  and  $p^B(x,y,t)$  are the pixels in last and present frames that fulfil with the IF condition will be taken in count to calculate the weighted mean,  $\alpha = \alpha_{SSS} = 0.1255$ .

<p><b>Fuzzy Rule <math>BBB(x,y,t)</math></b></p> <p>If <math>BBB(x,y,t)</math> is the biggest value found from the others:</p> <p><i>Step 1</i>) IF <math>\{(BBB(x,y,t) &gt; SBB(x,y,t)) \text{ AND } (BBB(x,y,t) &gt; SSS(x,y,t)) \text{ AND } (BBB(x,y,t) &gt; BBS(x,y,t))\}</math> THEN motion-noise = true.</p> $y_{out} = \frac{\sum p^A(x,y,t) \cdot SBB(x,y,t)}{\sum SBB(x,y,t)}$ <p><i>Step 2</i>) If <math>\sqrt{\text{motion-noise confidence}} = 1</math> . then <math>\alpha = 0.875</math> ,          else if <math>\sqrt{\text{motion-noise confidence}} = 0</math> then <math>\alpha = 0.125</math> , else <math>\alpha = 0.5</math> .</p> <p><i>Step 3</i>) <math>y_{out} = (1-\alpha) \cdot (pres\_fr_{central\_pixel}) + \alpha \cdot (past\_fr_{central\_pixel})</math></p>
--

The  $BBB(x,y,t)$  value shows that a central pixel and its neighbours have not relation among the others and it is highly probably that this pixel is in motion or is a noisy pixel. To solve this problem, consider the nine fuzzy gradient-vector values obtained from  $BBB(x,y,t)$  and take the central value and at least three fuzzy neighbours values more to detect movement present in the sample. We use the Fuzzy Rule “R” to obtain motion-noise confidence the activation degree of “R” is just the conjunction of the four subfacts, which are combined by a chosen triangular norm defined as  $A \text{ AND } B = A * B$  . Computations are specifically the intersection of all possible combinations of  $BBB(x,y,t)$  and three different neighbouring BIG membership degrees  $BBB(x+1,y+1,t)$ ,  $(i, j = -1, 0, 1)$ , using triangular norm. This can give 56 different values, which should be summed using algebraic sum  $A \text{ OR } B = A + B - A * B$  of all instances to obtain the motion-noise confidence.

<p><b>Fuzzy Rule <math>BBS(x,y,t)</math></b></p> <p>If <math>BBS(x,y,t)</math> is the biggest value found from the others:</p> <p><i>Step 1</i>) IF <math>\{(BBS(x,y,t) &gt; SBB(x,y,t)) \text{ AND } (BBS(x,y,t) &gt; SSS(x,y,t)) \text{ AND } (BBS(x,y,t) &gt; BBB(x,y,t))\}</math> THEN Weighted mean using <math>BBS(x,y,t)</math>,</p> $y_{out} = \frac{\sum p^B(x,y,t) \cdot (1-BBS(x,y,t))}{\sum (1-BBS(x,y,t))}$ <p><i>Step 2</i>) Update standard deviation for next frames to divide details from uniform regions.</p>
--

Now it can be applied the *Spatial Filter* to smooth the non-stationary noise left by the preceding temporal filter. This is done by a local spatial filter which adapts to image structures and noise levels present in the corresponding spatial neighbourhood.

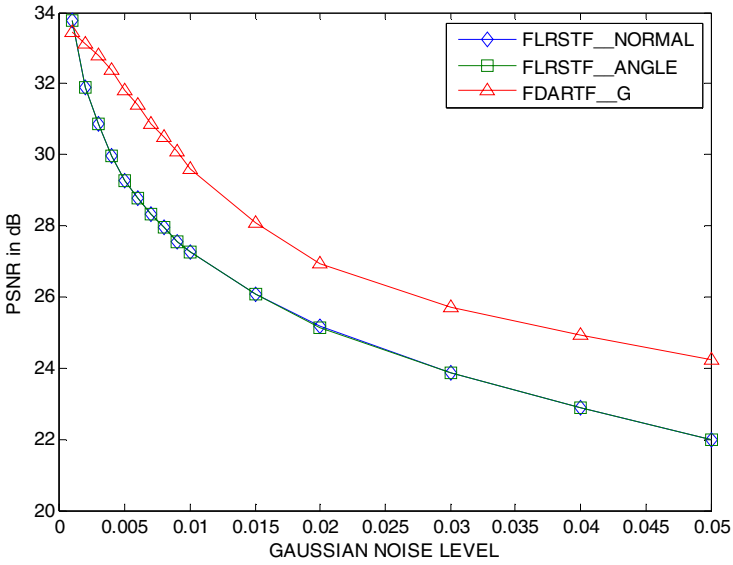


Fig. 3. PSNR values for a frame of video sequence "Miss America" by use different filters

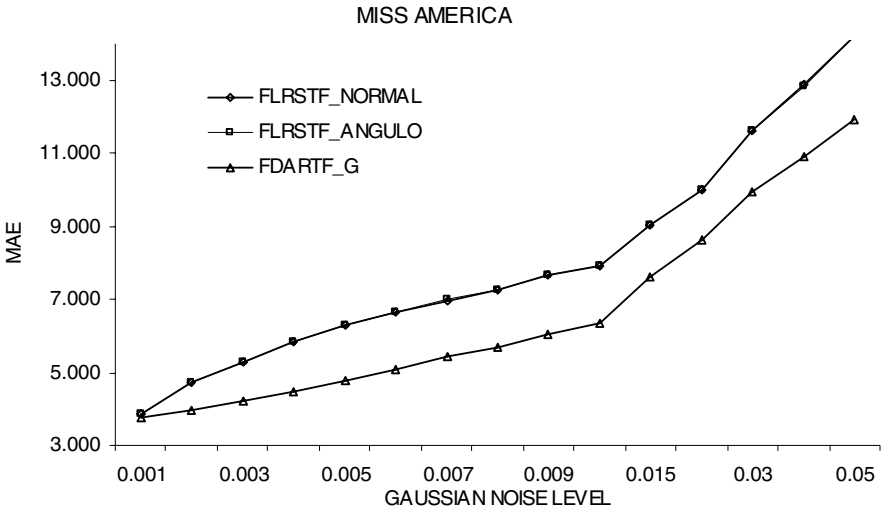


Fig. 4. MAE value for Miss America (Frame 100) with different Gaussian noise levels

### 3 Experimental Results

We obtained from the simulation experiments the properties of proposed Fuzzy Directional Adaptive Recursive Temporal (FDARTF\_G) filter and we compared it with the FMRSTF (Fuzzy Motion Recursive Spatial-Temporal) filter which works only with gradients [1-3], and with an adaptation to this algorithm using angle deviations FVMRSTF (Fuzzy Vectorial Motion Recursive Spatial-Temporal) filter which was not published yet.



**Fig. 5.** Visual results in a frame of video sequence “Flowers”, these images were restored from Gaussian noise corruption with variance 0.01, 0.02, and 0.03 from top to bottom, a) Column of restored images by FLRSTF\_NORMAL, and b) Column of restored images by proposed FDARTF\_G.



We use the video sequences “Flowers” and “Miss America” to qualify effectiveness of the proposed filter. The frames of video sequences are treated in an RGB color space with 24 bits, 8 bits for each channel, 176x144 pixels in a QCIF format with 100 frames.

Figure 3 presents the performance results in terms of PSNR for the frame #100 of video sequence “Miss America” corrupted with Gaussian noise from 0.00 to 0.05 in variance with zero mean by use different filters. From this Figure one can see that the best results in PSNR criterion are given by the proposed filter.

In Figure 4 we show the performance results in terms of MAE for the same frame of video sequence “Miss America”. This Figure we observe that the best results are given by the proposed filter.

Figure 5 depicts the visual results in a frame of video sequence “Flowers” by means of use of use the FLRSTF\_NORMAL and the proposed FDARTF\_G filter. These images were restored from Gaussian noise corrupted with variance 0.01, 0.02, and 0.03 from top to bottom of Figure 5. From this Figure, one can see that the restored frames by means of use the proposed filter appears to have a better subjective quality.

## 4 Conclusions

In this paper we present an adaptive recursive scheme for fuzzy logic based motion detection. The proposed algorithm realizes the spatial and temporal filtering to improve the noise suppression and detail preservation. We demonstrated that taking into account, both robust features (gradients and vectors) and connecting them together, we can realize a better algorithm, improving the techniques that use such features in a separate form. In future work, this method will be implemented to suppress impulsive random noise in video sequences.

## Acknowledgments

This work is supported by National Polytechnic Institute of Mexico and CONACyT.

## References

1. Zlokolica, V., De Geyter, M., Schulte, S., Pizurica, A., Philips, W., Kerre, E.: Fuzzy Logic Recursive Motion Detection for Tracking and Denoising of Video Sequences. IS&T/SPIE Symposium on Electronic Imaging, San Jose, California, USA, January (2005)
2. Zlokolica, V.: Advanced Non-Linear Methods for Video Denoising. PhD Thesis, Gent University (2006)
3. Zlokolica, V., Schulte, S., Pizurica, A., Philips, W., Kerre, E.: Fuzzy Logic Recursive Motion Detection and Denoising of Video Sequences. *Electronic Imaging* 15(2) (2006)
4. Trahanias, P.E., Venetsanopoulos, A.N.: Vector Directional Filters-A New Class of Multichannel Image Processing Filters. *IEEE Trans. Image Processing* 2(4) (1993)

5. Plataniotis, K.N., Venetsanopoulos, A.N.: *Color Image Processing and Applications*. Springer, Berlin (2000)
6. Schulte, S., De Witte, V., Nachtegaele, M., Van der Weken, D., Kerre, E.: Fuzzy Two-Step Filter for Impulse Noise Reduction from Color Images. *IEEE Trans. Image Processing* 15(11) (2006)
7. *Fuzzy Logic Fundamentals*, Ch. 3, pp. 61–103 (2001), [www.informit.com/content/images/0135705991/samplechapter/0135705991.pdf](http://www.informit.com/content/images/0135705991/samplechapter/0135705991.pdf)

# Bars Problem Solving - New Neural Network Method and Comparison\*

Václav Snášel<sup>1</sup>, Dušan Húšek<sup>2</sup>, Alexander Frolov<sup>3</sup>, Hana Řezanková<sup>4</sup>,  
Pavel Moravec<sup>1</sup>, and Pavel Polyakov<sup>5</sup>

<sup>1</sup> Department of Computer Science, FEEDS, VŠB – Technical University of Ostrava,  
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic  
{pavel.moravec, vaclav.snasel}@vsb.cz

<sup>2</sup> Institute of Computer Science, Dept. of Neural Networks, Academy of Sciences of Czech  
Republic, Pod Vodárenskou věží 2, 182 07 Prague, Czech Republic  
dusan@cs.cas.cz

<sup>3</sup> Institute of Higher Nervous Activity and Neurophysiology, Russian Academy of Sciences,  
Butlerova 5a, 117 485 Moscow, Russia  
aafrolov@mail.ru

<sup>4</sup> Department of Statistics and Probability, University of Economics, Prague, W. Churchill sq. 4,  
130 67 Prague, Czech Republic  
rezanka@vse.cz

<sup>5</sup> Institute of Optical Neural Technologies, Russian Academy of Sciences, Vavilova 44, 119 333  
Moscow, Russia  
pavel.mipt@mail.ru

**Abstract.** Bars problem is widely used as a benchmark for the class of feature extraction tasks. In this model, artificial data set is generated as a Boolean sum of a given number of bars. We show that the most suitable technique for feature set extraction in this case is neural network based Boolean factor analysis. Results are confronted with several dimension reduction techniques. These are singular value decomposition, semi-discrete decomposition and non-negative matrix factorization. Even if these methods are linear, it is interesting to compare them with neural network attempt, because they are well elaborated and are often used for a similar tasks. We show that frequently used cluster analysis methods can bring interesting results, at least for first insight to the data structure.

## 1 Introduction

In order to perform object recognition (no matter which one) it is necessary to learn representations of the underlying characteristic components. Such components correspond to objects, object-parts, or features. An image usually contains a number of different

---

\* The work was partly funded by the Center of Applied Cybernetics 1M6840070004 and partly by the Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Applications", by the project 1ET100300414 of the Program Information Society of the Thematic Program II of the National Research Program of the Czech Republic and by the project 201/05/0079 of the Grant Agency of the Czech Republic.

objects, parts, or features and these components can occur in different configurations to form many distinct images. Identifying the underlying components which are combined to form images is thus essential for learning the perceptual representations necessary for performing object recognition.

The feature extraction methods can use different aspects of images as the features, typically the color features (histograms), shape features (moments, contours, templates), texture features and others (e.g. eigenvectors). Such methods are either using a heuristics based on the known properties of the image collection, or are fully automatic and may use the original image vectors as an input. Here we will concentrate on the case of black and white pictures composed of  $32 \times 32$  pixels represented as binary vectors. Values of the entries of this vector represent individual pixels i.e. 0 for white and 1 for black. In this model, artificial data set is generated as a Boolean sum of pictures contained horizontal or vertical bars. These bars are features, we should find.

There exists many attempts that could be used for this reason. Here we will concentrate on the category which use dimension reduction techniques for automatic feature extraction.

First of all we will concentrate on the *neural network based Boolean factor analysis* developed by Frolov et al. [112]. Then we will compare the results with the most up to date linear procedures.

One of the most popular methods today is the *singular value decomposition* which was already many times successfully used for the automatic feature extraction. In case of bar collection, base vectors can be interpreted as images, describing some common characteristics of several input signals. These base vectors are often called eigenfaces in the case of face recognition task, see Turk et al. [3] and [89,10].

However singular value decomposition is not suitable for huge collections and is computationally expensive, so other methods of dimension reduction were proposed. Here we apply *semi-discrete decomposition*.

Because the data matrix does not have all elements non-negative, we tried to apply a new method, called *non-negative matrix factorization* as well.

However not only for the first view on data structures, one can apply traditional statistical methods, mainly different algorithms for statistical cluster analysis.

The rest of this paper is organized as follows. The second section explains the dimension reduction methods used in this study. Then in the section three we describe experimental results, and finally in the section four there are some conclusions.

## 2 Dimension Reduction

We used a new *Neural network based Boolean Factor Analysis (NBFA)* first and then three other promising methods of dimension reduction for our comparison *Singular Value Decomposition (SVD)*, *Semi-Discrete Decomposition (SDD)*, and *Non-negative Matrix Factorization (NMF)*. From classical statistical clustering methods, we applied *Hierarchical Agglomerative Algorithm (HAA)*, and *Two-Step Cluster Analysis (TSCA)*. All of them are briefly described below.

### 2.1 Neural Network Based Boolean Factor Analysis

The *NBFA* method is a powerful tool for revealing the information redundancy of high dimensional binary signals [2]. It allows to express every signal (vector of variables values) from binary data matrix  $X$  of observations as superposition of binary factors:

$$X = \bigvee_{l=1}^L S_l f^l, \tag{1}$$

where  $S_l$  is a component of factor scores and  $f^l$  is a vector of factor loadings and  $\vee$  denotes Boolean summation ( $0 \vee 0 = 0, 1 \vee 0 = 1, 0 \vee 1 = 1, 1 \vee 1 = 1$ ). If we mark Boolean matrix multiplication by the symbol  $\odot$ , then we can express approximation of data matrix  $X$  in matrix notation

$$X \simeq F \odot S \tag{2}$$

where  $S$  is the matrix of factor scores and  $F$  is the matrix of factor loadings. The Boolean factor analysis implies that components of original signals, factor loadings and factor scores are binary values.

Optimal solution of  $X$  decomposition according [2] by brute force search is NP-hard problem and as such is not suitable for high dimensional data. On other side the classical linear methods could not take into account non-linearity of Boolean summation and therefore are inadequate for this task.

The *NBFA* is based on Hopfield-like neural network [4][1]. Used is the fully connected network of  $N$  neurons with binary activity (1 - active, 0 - nonactive). Each pattern of the learning set  $X^m$  is stored in the matrix of synaptic connections  $J'$  according to Hebbian rule:

$$J'_{ij} = \sum_{m=1}^M (X_i^m - q^m)(X_j^m - q^m), \quad i, j = 1, \dots, N, \quad i \neq j, \quad J'_{ii} = 0 \tag{3}$$

where  $M$  is the number of patterns in the learning set and bias  $q^m = \sum_{i=1}^N X_i^m / N$  is the total relative activity of the  $m$ -th pattern. This form of bias corresponds to the biologically plausible global inhibition being proportional to overall neuron activity. One special inhibitory neuron was added to  $N$  principal neurons of the Hopfield network. The neuron was activated during the presentation of every pattern of the learning set and was connected with all the principal neurons by bidirectional connections. Patterns of the learning set are stored in the vector  $J''$  of the connections according to the Hebbian rule:

$$J''_i = \sum_{m=1}^M (X_i^m - q^m) = M(q_i - q), \quad i = 1, \dots, N, \tag{4}$$

where  $q_i = \sum_{m=1}^M X_i^m / M$  is a mean activity of the  $i$ -th neuron in the learning set and  $q$  is a mean activity of all neurons in the learning set. We also supposed that the excitability

of the introduced inhibitory neuron decreases inversely proportional to the size of the learning set, being  $1/M$  after all patterns are stored. In the recall stage its activity is then:

$$A(t) = (1/M) \sum_{i=1}^N J''_i X_i(t) = (1/M) J''^T X(t)$$

where  $J''^T$  is transposed  $J''$ . The inhibition produced in all principal neurons of the network is given by vector  $J'' A(t) = (1/M) J'' J''^T X(t)$ . Thus, the inhibition is equivalent to the subtraction of

$$J'' = J'' J''^T / M = M Q Q^T \tag{5}$$

from  $J'$  where  $Q$  is a vector with components  $q_i - q$ . Adding the inhibitory neuron is equivalent to replacing the ordinary connection matrix  $J'$  by the matrix  $J = J' - J''$ .

To reveal factors we suggest the following two-run recall procedure. Its initialization starts by the presentation of a random initial pattern  $X^{in}$  with  $k_{in} = r_{in} N$  active neurons. Activity  $k_{in}$  is supposed to be smaller than the activity of any factor. On presentation of  $X^{in}$ , network activity  $X$  evolves to some attractor. This evolution is determined by the synchronous discrete time dynamics. At each time step:

$$X_i(t + 1) = \Theta(h_i(t) - T(t)), \quad i = 1, \dots, N, \quad X_i(0) = X_i^{in} \tag{6}$$

where  $h_i$  are components of the vector of synaptic excitations

$$h(t) = J X(t), \tag{7}$$

$\Theta$  is the step function, and  $T(t)$  is an activation threshold.

At each time step of the recall process the threshold  $T(t)$  was chosen in such a way that the level of the network activity was kept constant and equal to  $k_{in}$ . Thus, on each time step  $k_{in}$  “winners” (neurons with the greatest synaptic excitation) were chosen and only they were active on the next time step. As shown in [4], this choice of activation threshold enables the network activity to stabilize in point or cyclic attractors of length two. The fixed level of activity at this stage of the recall process could be ensured by biologically plausible non-linear negative feed-back control accomplished by the inhibitory interneurons. It is worth to note that although the fact of convergence of network synchronous dynamics to point or cyclic attractors of length two was established earlier [5], it was done for fixed activation threshold  $T$  but for fixed network activity it was done first in [4].

When activity stabilizes at the initial level of activity  $k_{in}$ ,  $k_{in} + 1$  neurons with maximal synaptic excitation are chosen for the next iteration step, and network activity evolves to an attractor at the new level of activity  $k_{in} + 1$ . The level of activity then increases to  $k_{in} + 2$ , and so on, until the number of active neurons reaches the final level  $k_f = r_f N$  where  $r = k/N$  is a relative network activity. Thus, one trial of the recall procedure consists of  $(r_f - r_{in})N$  external steps and several internal steps (usually 2-3) inside each external step to reach an attractor for a given level of activity.

At the end of each external step when network activity stabilizes at the level of  $k$  active neurons a Lyapunov function was calculated by formula:

$$\lambda = X^T(t + 1)JX(t)/k, \tag{8}$$

where  $X^T(t+1)$  and  $X(t)$  are two network states in a cyclic attractor (for point attractor  $X^T(t + 1) = X(t)$ ). The identification of factors along the trajectories of the network dynamics was based on the analysis of the change of the Lyapunov function and the activation threshold along each trajectory. In our definition of Lyapunov function its value gives a mean synaptic excitation of neurons belonging to an attractor at the end of each external step.

### 2.2 Singular Value Decomposition

The *SVD*, see Berry et al. [6], is an algebraic extension of classical vector model. It is similar to the *Principal Component Analysis PCA* method, which was originally used for the generation of eigenfaces. Informally, *SVD* discovers significant properties and represents the images as linear combinations of the base vectors. Moreover, the base vectors are ordered according to their significance for the reconstructed image, which allows us to consider only the first  $k$  base vectors as important (the remaining ones are interpreted as "noise" and discarded). Furthermore, *SVD* is often referred to as more successful in recall when compared to querying whole image vectors see [6] again. Formally, we decompose the matrix of images  $A$  by singular value decomposition, calculating singular values and singular vectors of  $A$ .

We have matrix  $A$ , which is an  $n \times m$  rank- $r$  matrix (where  $m \geq n$  without loss of generality) and values  $\sigma_1, \dots, \sigma_r$  are calculated from eigenvalues of matrix  $AA^T$  as  $\sigma_i = \sqrt{\lambda_i}$ . Based on them, we can calculate column-orthonormal matrices  $U = (u_1, \dots, u_n)$  and  $V = (v_1, \dots, v_m)$ , where  $U^T U = I_n$  and  $V^T V = I_m$ , and a diagonal matrix  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ , where  $\sigma_i > 0$  for  $i \leq r$ ,  $\sigma_i \geq \sigma_{i+1}$  and  $\sigma_{r+1} = \dots = \sigma_n = 0$ .

The decomposition

$$A = U \Sigma V^T \tag{9}$$

is called *singular decomposition* of matrix  $A$  and the numbers  $\sigma_1, \dots, \sigma_r$  are *singular values* of the matrix  $A$ . Columns of  $U$  (or  $V$ ) are called *left* (or *right*) singular vectors of matrix  $A$ .

Now we have a decomposition of the original matrix of images  $A$ . We get  $r$  nonzero singular numbers, where  $r$  is the rank of the original matrix  $A$ . Because the singular values usually fall quickly, we can take only  $k$  greatest singular values with the corresponding singular vector coordinates and create a *k-reduced singular decomposition* of  $A$ . Let us have  $k$  ( $0 < k < r$ ) and singular value decomposition of  $A$

$$A = U \Sigma V^T \approx A_k = (U_k U_0) \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^T \\ V_0^T \end{pmatrix} \tag{10}$$

We call  $A_k = U_k \Sigma_k V_k^T$  a *k-reduced singular value decomposition (rank- $k$  SVD)*. Instead of the  $A_k$  matrix, a matrix of image vectors in reduced space  $D_k = \Sigma_k V_k^T$  is used

in *SVD* as the representation of image collection. The image vectors (columns in  $D_k$ ) are now represented as points in  $k$ -dimensional space (the *feature-space*) represent the matrices  $U_k, \Sigma_k, V_k^T$ .

Rank- $k$  *SVD* is the best rank- $k$  approximation of the original matrix  $A$ . This means that any other decomposition will increase the approximation error, calculated as a sum of squares (*Frobenius norm*) of error matrix  $B = A - A_k$ . However, it does not implicate that we could not obtain better precision and recall values with a different approximation.

Once computed, *SVD* reflects only the decomposition of original matrix of images. If several hundreds of images have to be added to existing decomposition (*folding-in*), the decomposition may become inaccurate. Because the recalculation of *SVD* is expensive, so it is impossible to recalculate *SVD* every time images are inserted. The *SVD-Updating* is a partial solution, but since the error slightly increases with inserted images. If the updates happen frequently, the recalculation of *SVD* may be needed soon or later.

### 2.3 Semi-discrete Decomposition

The *SDD* method is one of other *SVD* based methods, proposed recently for text retrieval in Kolda et al. [7]). As mentioned earlier, the rank- $k$  *SVD* method (called *truncated SVD* by authors of semi-discrete decomposition) produces dense matrices  $U$  and  $V$ , so the resulting required storage may be even larger than the one needed by the original term-by-document matrix  $A$ . To improve the required storage size and query time, the semi-discrete decomposition was defined as

$$A \approx A_k = X_k D_k Y_k^T, \quad (11)$$

where each coordinate of  $X_k$  and  $Y_k$  is constrained to have entries from the set  $\varphi = \{-1, 0, 1\}$ , and the matrix  $D_k$  is a diagonal matrix with positive coordinates.

The *SDD* method does not reproduce  $A$  exactly, even if  $k = n$ , but it uses very little storage with respect to the observed accuracy of the approximation. A rank- $k$  *SDD* (although from mathematical standpoint it is a sum on rank-1 matrices) requires the storage of  $k(m+n)$  values from the set  $\{-1, 0, 1\}$  and  $k$  scalars. The scalars need to be only single precision because the algorithm is self-correcting. The *SDD* approximation is formed iteratively. The optimal choice of the triplets  $(x_i, d_i, y_i)$  for given  $k$  can be determined using greedy algorithm, based on the residual  $R_k = A - A_{k-1}$  (where  $A_0$  is a zero matrix).

### 2.4 Non-negative Matrix Factorization

The *NMF* method calculates an approximation of the matrix  $A$  as a product of two matrices,  $W$  and  $H$ . The matrices are usually pre-filled with random values (or  $H$  is initialized to zero and  $W$  is randomly generated). During the calculation the values in  $W$  and  $H$  stay positive. The approximation of matrix  $A$ , matrix  $A_k$ , can be calculated



as  $A_k = WH$ . The original *NMF* method tries to minimize the Frobenius norm of the difference between  $A$  and  $A'_k$  using

$$\min_{W,H} \|V - WH\|_F^2 \tag{12}$$

as the criterion in the minimization problem. Recently, a new method was proposed in (M. W. Spratling [11]), where the constrained least squares problem is solved

$$\min_{H_j} \{ \|V_j - WH_j\|_2^2 - \lambda \|H_j\|_2^2 \} \tag{13}$$

as the criterion in the minimization problem. This approach yields better results for sparse matrices. Here, unlike in *SVD*, the base vectors are not ordered from the most general one and we have to calculate the decomposition for each value of  $k$  separately.

### 2.5 Statistical Clustering Methods

To set our method in more global context we applied cluster analysis, too. The clustering methods help to reveal groups of similar features, it means typical parts of images. However, obtaining disjunctive clusters is a problem of the usage of traditional hard clustering. We have chosen two most promising techniques available in recent statistical packages – hierarchical agglomerative algorithm and two-step cluster analysis.

The *HAA* algorithm starts with each feature in a group of its own. Then it merges clusters until only one large cluster remains which includes all features. The user must choose dissimilarity or similarity measure and agglomerative procedure. At the first step, when each feature represents its own cluster, the dissimilarity between two features is defined by the chosen dissimilarity measure. However, once several features have been linked together, we need a linkage or amalgamation rule to determine when two clusters are sufficiently similar to be linked together. Several linkage rules have been proposed.

For example, the distance between two clusters can be determined by the greatest distance between two features from the different clusters (*complete linkage method – CL*), or average distance between all pairs of features from two different clusters (*average linkage between groups – ALBG*). Hierarchical clustering is based on the proximity matrix (entries are dissimilarities for all pairs of features). For binary data, we can be used Jaccard or Ochiai (cosine) similarity measure for example.

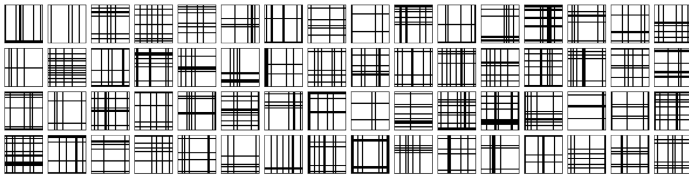
The former can be expressed as  $S_J = \frac{a}{a+b+c}$  where  $a$  is the number of the common occurrences of ones and  $b + c$  is the number of pairs in which one value is one and the second is zero. The latter can be expressed as  $S_O = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$ . The further clustering techniques mentioned above are suitable for large data files but they are independent on the order of features.

In two-step cluster analysis (*TSCA*), the features are arranged into sub-clusters, known as cluster features (CF), first. These cluster features are then clustered into  $k$  groups, using a traditional hierarchical clustering procedure. A cluster feature represents a set of summary statistics on a subset of the data. The algorithm consists of two phases. In the first one, an initial CF tree is built (a multi-level compression of the data

that tries to preserve the inherent clustering structure of the data). In the second one, an arbitrary clustering algorithm is used to cluster the leaf nodes of the CF tree. Advantage of this method is its ability to work with larger data sets; disadvantage then, is its sensitivity to the order of the features. In the implementation in the SPSS system, the log-likelihood distance is applied.

### 3 Experimental Results

For testing of above mentioned methods, we used generic collection of 1600  $32 \times 32$  black-and-white images containing different combinations of horizontal and vertical lines (bars). The probabilities of bars to occur in images were the same and equal to  $10/64$ , i.e. images contain 10 bars in average. An example of several images from generated collection is depicted in Figure 1.



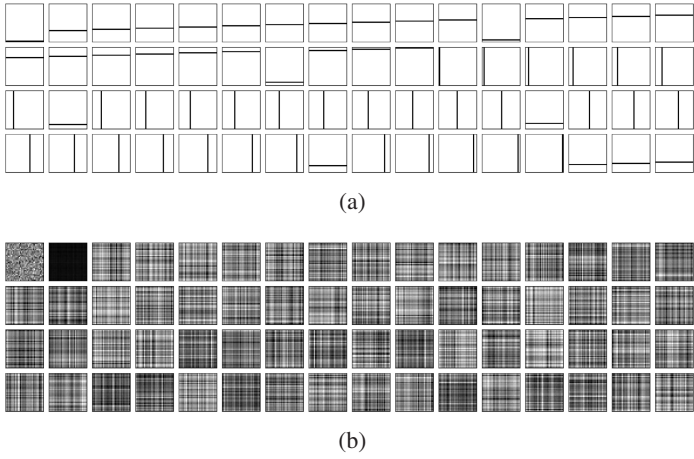
**Fig. 1.** Several images from generated collection

First, we made decomposition of images into binary vectors by *NBFA* method. Here factors contains only values  $\{0, 1\}$  and model is Boolean. The factor search was performed under assumption that the number of ones in factor is not less than 5 and not greater than 200. Since the images are obtained by Boolean summation of binary bars, it is not surprising, that *NBFA* is able to reconstruct all bars as factors, providing an ideal solution, as we can see in Figure 2a.

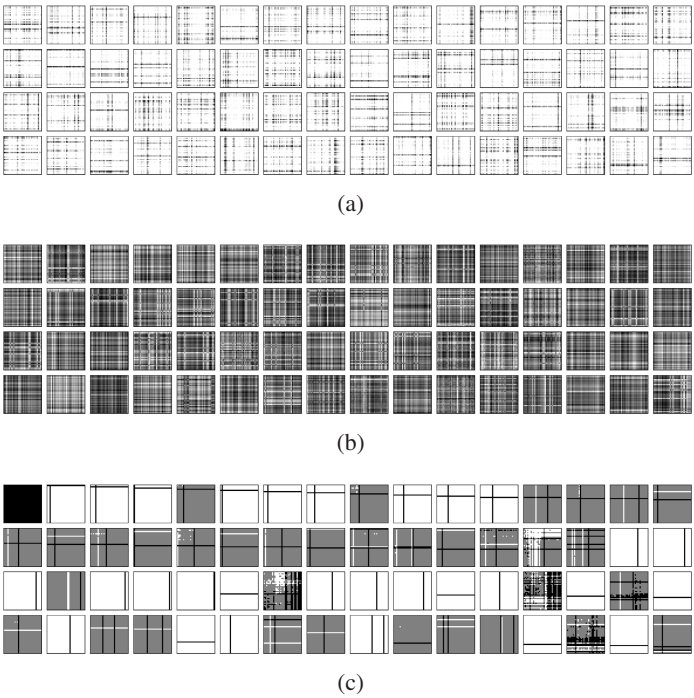
It is clear that classical linear methods could not take into account non-linearity of Boolean summation and therefore are inadequate for Bar problem task. But the linear methods are fast and well elaborated so it was very interesting to compare linear approach with the *NBFA* and compare the methods at least qualitatively.

With *SVD* method, we obtain singular vectors, the most general being among the first. The first few are shown in Figure 2b. We can see, that the bars are not separated and different shades of gray appear.

The *NMF* methods yield different results. The original *NMF* method, based on the adjustment of random matrices  $W$  and  $H$  provides hardly-recognizable images even for  $k = 100$  and 1000 iterations (we used 100 iterations for other experiments). Moreover, these base images still contain significant salt and pepper noise and have a bad contrast. The factors are shown in Figure 3a. We must also note, that the *NMF* decomposition will yield slightly different results each time it is run, because the matrix(es) are pre-filled with random values.



**Fig. 2.** First 64 factors retrieved by *NBFA* method (a) First 64 factors (singular vectors) computed by original *SVD* method (b)



**Fig. 3.** First 64 factors (base images) calculated by *NMF* method (a) First 64 factors calculated by *GD-CLS NMF* method (b) First 64 factors (base images) calculated by *SDD* method First 64 factors (base vectors) for *SDD* method (c)

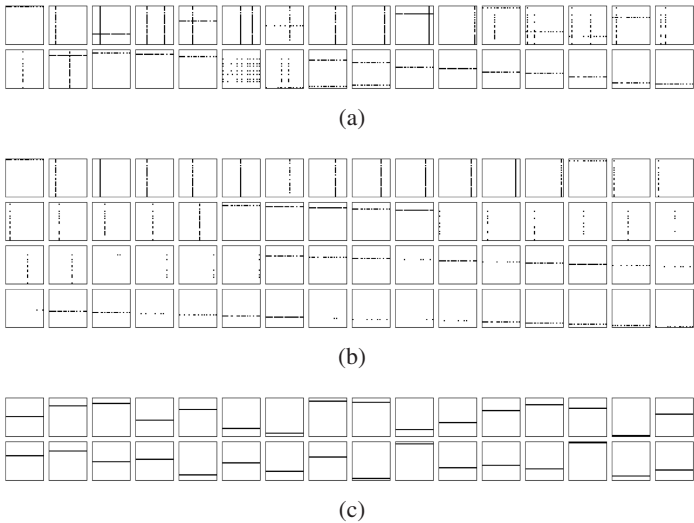
The *GD-CLS* modification of *NMF* method (proposed in Shahnaz et al. [12]) tries to improve the decomposition by calculating the constrained least squares problem. This leads to a better overall quality, however, the decomposition really depends on the pre-filled random matrix  $H$ . The result is shown in Figure 3b.

The *SDD* method differs slightly from previous methods, since each factor contains only values  $\{-1, 0, 1\}$ . Gray in the factors shown in Figure 3c represents 0;  $-1$  and  $1$  are represented with black and white respectively. The base vectors in Figure 3c can be divided into three categories: Base vectors containing only one bar. Base vectors containing one horizontal and one vertical bar. Other base vectors, containing several bars and in some cases even noise.

At the end we applied traditional cluster analysis. We clustered 1024 ( $32 \times 32$ ) positions into 64 and 32 clusters. The problem of the use of traditional cluster analysis consists in that we can obtain disjunctive clusters only. So we can find only horizontal bars and parts of vertical bars and vice versa.

For or testing we used *HAA* and then *TSCA* clustering as implemented in the *SPSS* system. The problem of the last type of the analysis consists in that it is dependent on the order of analyzed images. So we used two different orders; in the second one we swapped images from 1001 to 1009 with images 100 and 101.

For *HAA*, we tried to use different similarity measures. We found that linkage methods have more influence to the results of clustering than similarity measures. We used Jaccard and Ochiai (cosine) similarity measures suitable for asymmetric binary attributes. We found both as suitable methods for the identification of the bars or their parts. For 64 clusters, the differences were only in a few assignments of positions by *ALBG* and *CL* methods with Jaccard and Ochiai measures.



**Fig. 4.** 32 clusters of pixels by *ALBG* method (Jaccard coefficient) (a) 64 clusters of pixels by *ALBG* method (Jaccard coefficient) (b) 32 clusters of pixels by *TSCA* method (c)

Above shown figures illustrate the application of some of these techniques for 64 and 32 clusters. Figures 4 a,b,c show results of *ALBG* method with Jaccard measure Figure 4 a for 32 clusters and Figure 4 b for 64 clusters. In the case of 32 clusters, we found 32 horizontal bars (see Figure 4 c ) by *TSCA* method for the second order of features.

## 4 Conclusion

In this paper, we have compared several dimension reduction and clustering methods on bars collection. the *NBFA* perfectly found basis (factors) from which the whole set of learned pictures can be reconstructed. First, it is because the model, on which the *NBFA* is based, is the same as used for data generation. Secondly, because of robustness of *NBFA* implementation based on recurrent neural network. Some experiments show, that the resistance against noise is very high. We hypothesize that it is due the self reconstruction ability of our neural network. The *SDD* method is tree valued so its answers are very often "I do not know".

The *SVD* and *NMF* methods yield slightly worse results, since they are not focused on binary data. The *NMF* methods are restricted on positive values, but the results are still not as good as in the case of *NBFA*.

Cluster analysis is focused on finding original factors from which images were generated. Applied clustering methods were quite successful in finding these factors even if they are capable, it follows from their principle, to find only disjunctive clusters. So, only some bars or their parts were revealed. However, from the general view on 64 clusters, it is obvious, that images are compounded from vertical and horizontal bars (lines). By two-step cluster analysis 32 horizontal lines were revealed by clustering to 32 clusters.

## References

1. Frolov, A.A., Sirota, A.M., Húsek, D., Muravjev, P.: Binary factorization in Hopfield-like neural networks: single-step approximation and computer simulations. *Neural Networks World*, 139–152 (2004)
2. Frolov, A.A., Húsek, D., Muravjev, P., Polyakov, P.: Boolean Factor Analysis by Attractor Neural Network. *Neural Networks, IEEE Transactions* 18(3), 698–707 (2007)
3. Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3(1), 71–86 (1991)
4. Frolov, A.A., Húsek, D., Muravjev, P.: Informational efficiency of sparsely encoded Hopfield-like autoassociative memory. In: *Optical Memory and Neural Networks (Information Optics)*, pp. 177–198 (2003)
5. Goles-Chacc, E., Fogelman-Soulie, F.: Decreasing energy functions as a tool for studying threshold networks. *Discrete Mathematics*, 261–277 (1985)
6. Berry, M., Dumais, S., Letsche, T.: Computational Methods for Intelligent Information Access. In: *Proceedings of the 1995 ACM/IEEE Supercomputing Conference*, San Diego, California, USA (1995)
7. Kolda, T.G., O'Leary, D.P.: Computation and uses of the semidiscrete matrix decomposition. In: *ACM Transactions on Information Processing*, pp. 415–435. ACM Press, New York (2000)

8. Moravec, P., Snášel, V.: Dimension Reduction Methods for Image Retrieval Intelligent Systems Design and Applications, 2006. In: ISDA 2006. Sixth International Conference on Intelligent Systems Design and Applications, vol. 2, pp. 1055–1060 (October 2006)
9. Praks, P., Dvorský, J., Snášel, V.: Latent semantic indexing for image retrieval systems. In: Proceedings of the SIAM Conference on Applied Linear Algebra, Williamsburg (2003)
10. Praks, P., Machala, L., Snášel, V.: Iris Recognition Using the SVD-Free Latent Semantic Indexing. In: Khan, L., Petrushin, V.A. (eds.) (MDM/KDD 2004). Proceeding of the Fifth International Workshop on Multimedia Data Mining, pp. 67–71. Springer, Heidelberg (2006)
11. Spratling, M.W.: Learning Image Components for Object Recognition. *Journal of Machine Learning Research* 7, 793–815 (2006)
12. Shahnaz, F., Berry, M., Pauca, P., Plemmons, R.: Document clustering using nonnegative matrix factorization. *Journal on Information Processing and Management* 42, 373–386 (2006)

# A Coarse-and-Fine Bayesian Belief Propagation for Correspondence Problems in Computer Vision

Preeyakorn Tipwai<sup>1</sup> and Suthep Madarasmi<sup>2</sup>

<sup>1</sup> Rajamangala University of Technology Lanna, Chiangmai 50300, Thailand  
preeyakorn@rmutl.ac.th

<sup>2</sup> King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand  
suthep@kmutt.ac.th

**Abstract.** We present the use of a multi-resolution, coarse-and-fine, pyramid image architecture to solve correspondence problems in various computer vision modules including shape recognition through contour matching, stereovision, and motion estimation. The algorithm works with a grid matching and an inter-grid correspondence model by message passing in a Bayesian belief propagation (BBP) network. The local smoothness and other constraints are expressed within each resolution scale grid and also between grids in a single paradigm. Top-down and bottom-up matching are concurrently performed for each pair of adjacent levels of the image pyramid level in order to find the best matched features at each level simultaneously. The coarse-and-fine algorithm uses matching results in each layer to constrain the process in its 2 adjacent upper and lower layers by measuring the consistency between corresponding points among adjacent layers so that good matches at different resolution scales constrain one another. The coarse-and-fine method helps avoid the local minimum problem by bringing features closer at the coarse level and yet providing a complete solution at the finer level. The method is used to constrain the solution with examples in shape retrieval, stereovision, and motion estimation to demonstrate its desirable properties such as rapid convergence, the ability to obtain near optimal solution while avoiding local minima, and immunity to error propagation found in the coarse-to-fine approach. . . .

## 1 Introduction

Computer vision problems are often formulated as an optimization approach involving energy minimization. This method is used for estimating some spatially varying parameter such as orientation or depth from observed noisy data. Horn and Schunk [1] introduced a gradient descent approach that is phrased in continuous terms by applying update rules defined by the Euler equations to each pixel. It has difficulty in obtaining update rules for complex energy functions, and guaranteed convergence takes too long.

Discrete relaxation methods became popular in later works. Iterated Conditional Modes (ICM) was instituted by Besag [2]. In their work, the variation of

energy is explored along a space of all possible values at each pixel, and the value that gives the largest decrease of the total energy is chosen. The process often results in a local minimum that is very far from the optimum, on account of the algorithm's greedy nature. Geman and Geman [3] introduced an approach using the Gibbs Sampler with simulated annealing. It aimed at incorporating observational information by regarding the labeling task as finding the maximum a posteriori probability estimation. Minimizing an arbitrary energy functional requires exponential time, making simulated annealing intolerably slow. Bilbro et al. [4] introduced Mean Field Annealing (MFA) which replaces random search in simulated annealing with a series of deterministic gradient descent to avoid the long computational time problem. However, the task of computing the partition function required for MFA is often computationally intractable. Amini et al. [5] presented an approach to minimize the energy function via a "time-delayed" discrete dynamic programming algorithm on the problems set up as a discrete multistage decision process. As a specific example, they applied the dynamic programming energy-minimizing method to active contours. Nevertheless, there appears a limitation to dynamic programming; namely, the restriction on one-dimensional setting of the energy functions.

Energy minimization via graph cuts presented by Boykov et al. [6] has been a popular optimization method for solving vision problems. By considering a broad category of energy functions using a variety of smoothness and other prior constraints, the graph cuts algorithm finds a local minimum by performing two types of large moves. One type is called the expansion move, which works by finding a labeling within a known factor of the global minimum. The other type is the swap move, which deals with more ordinary energy functions. Both types of moves make it possible to update a large number of pixel labeling concurrently, in contrast with the most traditional algorithms which tend to change the value of only one pixel at a time.

Bayesian belief propagation (BBP) was first invented by Pearl [7] in 1986. His algorithm consists of executing simple local belief updates to pass the message along the hidden nodes of a Markov Random Field (MRF), and compute the belief revision to find the most likely sequence. Pearl showed that the algorithm works well for singly connected networks. Weiss [8] proposed a belief propagation and revision in networks with loops, and then applied the loopy belief propagation to image interpretation in [9]. Since then, a number of researchers have presented several works on using belief propagation in computer vision modules.

BBP algorithms for image segmentation have been proposed such as the work by Shental et. al [10] called the generalized belief propagation typical cut used for learning and inferring segmentations. Kolmogorov [11] applied the Tree Reweighted Belief Propagation (TRBP) to the stereopsis problem. Sun et. al [12] proposed a variety of BBP for stereo matching which even included a line process for depth discontinuity and a binary process to handle occlusions. Coughland et al. [13] proposed the use of the BBP to find selected feature points of a template contour in the target image constrained by a relationship similarity of corresponding neighborhood pairs in the template and in the target contours.



Gao et al. [14] proposed the use of a multi-frame belief propagation for Bayesian inference in a graph model which infers human motion and posed the motion inference as a mapping problem between state nodes in the graph.

Multi-resolution image representation and processing is a wellknown image analysis methodology with the goal of faster and more robust performance. Many researches have employed multi-resolution representations to solve vision problems. A popular multi-resolution strategy is the coarse-to-fine approach. The process starts at a very coarse resolution level, where the size of the data representation is small compared to the full resolution input image. Matching results at each resolution level are then used to guide the matching at the next, higher resolution level. The coarse-to-fine approach is a commonly used architecture in many vision algorithms. For example, Marapane et al. [15] proposed a multi-primitive hierarchical method (MPH), which used a hierarchy of primitives of different abstractions, with higher levels using richer primitives for stereo analysis. Jain et al. [16] uses a coarse-to-fine strategy for deformable templates to find a match between a sketched template and various target images.

Felzenszwalb and Huttenlocher [17] proposed a coarse-to-fine BBP approach with an added approximation to reduce the computational complexity inherent in the classical BBP algorithm. The data cost constraint is precalculated on the original resolution image, then sub-sampled into several coarser levels. The message passing among neighborhood pixels are calculated firstly in the coarsest level for a certain number of iterations, and then copied to the finer level. The messages are then calculated in each level and copied to the finer level in the same way until the process at the finest level is complete. At the end the belief is computed at the final, full-scale level.

Problems of the coarse-to-fine approach exist due to error propagation from the course to the fine solution. This error is partly due to the loss of some information caused by subsampling images at the coarser levels. If an error occurs at an early stage, this error in the lower resolution image is propagated into each subsequent higher resolution level, worsening the solution. This error cannot be corrected by using information at any level because the information flows in a top-down, feedforward manner without feedback from higher resolution levels.

To deal with error propagation in the coarse-to-fine approach, we propose a coarse-and-fine approach to belief propagation by passing messages across grid levels, in addition to those within the same grid. We pass messages in both directions between grids: both from coarse to fine and from fine back to coarse. Since the solution at a coarser scale should constrain the solutions at its adjacent finer scale level while at the same time the finer scale solution should also constrain the solution at the corresponding, adjacent coarser level, we use this information to tightly couple the 2 systems. The computations are concurrently performed in order to find the best matched feature of all levels simultaneously. Matching results in each layer are used to constrain the process in adjacent layers by passing a message containing the inter-grid matching constraint to measure the consistency of solutions between adjacent image resolutions. Each level implements a matching process between two features using the same resolution level. At the

same time, each level communicates with adjacent resolution levels permitting interlayer feedback during the matching process. Hence, good matches at multiple levels constrain one another. A more accurate label map can be obtained by using the multi-resolution images in parallel. The coarse-and-fine method avoids the local minimum problem by bringing features closer at the coarse level and yet providing a complete solution at the finer level.

## 2 The Coarse-and-Fine Belief Propagation

The structure of our coarse-and-fine BBP is illustrated in Fig. 1. Let  $I^0$  be the input observed image, which is sub-sampled into  $N - 1$  coarser levels  $I^n, n = 1, \dots, N - 1$ . For stereovision  $I$  may be 2 images  $I_{left}$  and  $I_{right}$ . Let  $f^n = f_i^n$  be the scene property such as disparity at grid level  $n$ . Within each grid the data constraint is computed through differences in matching points and the smoothness constraint in the computed solution. Both the data and smoothness constraints are enforced within a single resolution to find the disparity labeling in that scale. In addition, the inter-grid constraint requires that the computed disparity at the neighboring coarse and fine levels must be consistent. Message passing in the coarse-and-fine structure is illustrated in Fig. 2. The grid matching is in the messages  $m_{left}, m_{right}, m_{up}, m_{down}$ , which are sent to neighbors in the same grid level for the smoothness constraint. The inter-grid consistency constraint is enforced by the message  $m_{above}$  sent to corresponding pixels in the upper scale level, and the messages  $m_{below1}, m_{below2}, m_{below3}, m_{below4}$  sent to four corresponding pixels in the lower scale level.

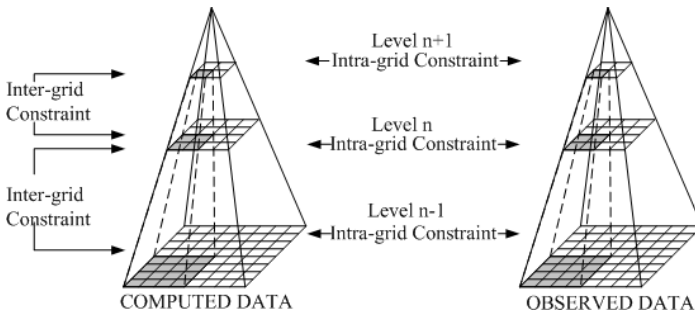


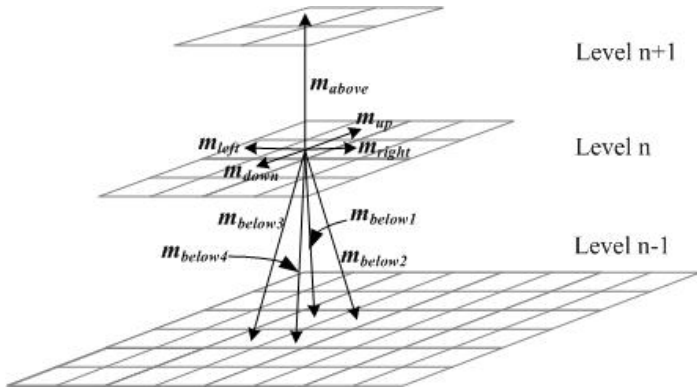
Fig. 1. The architecture for the coarse-and-fine pyramid

### 2.1 Computing Local Evidence

Given multiple resolution images  $I^n, n = 1, \dots, N - 1$ , the local evidence or data constraint at each pixel  $p$  on level  $n$  can be computed via:

$$m_p^n(f_p^n) = \ell^{-\lambda_D E_p^{DATA(n)}(f_p^n)} \tag{1}$$

where  $E_p^{DATA(n)}(f_p^n)$  is the data cost function at pixel  $p$  having the value  $f_p^n$ . The computational method depends on the application.



**Fig. 2.** Message passing in coarse-and-fine belief propagation. Messages  $m_{left}, m_{right}, m_{up}, m_{down}$  enforce the intra-grid constraints. Messages  $m_{above}, m_{below1}, m_{below2}, m_{below3}, m_{below4}$  enforce the inter-grid constraints.

### 2.2 Computing Intra-grid Message

To send a message at the same grid level, the message is computed by

$$m_{pq}^n(f_q^n) \leftarrow \max_{f_p^n} \ell^{-\lambda_S E_{pq}^{SMOOTH}^{(n)}(f_p^n, f_q^n)} m_p^n(f_p^n) \prod_{r \in N(p) \setminus q} m_{rp}^n(f_r^n) \quad (2)$$

where  $p = (x, y)$ ,  $m_{pq}$  is  $m_{left}, m_{right}, m_{up}, m_{down}$  when  $q = (x + 1, y), (x - 1, y), (x, y + 1)$ , and  $(x, y - 1)$ , respectively,  $E_{pq}^{SMOOTH}^{(n)}(f_p^n, f_q^n)$  is the smoothness energy between nodes  $p$  and  $q$ , which are both at the same level  $n$ . Thus, when node  $p$  sends a message to a neighbor  $q$ , it will sum up the evidence from observed data (difference in intensity between matched left and right images), the smoothness energy, and all the messages it receives from all other intra-grid neighbors,  $m_{left, right, up, down}$ , and from all inter-grid neighbors  $m_{above}, m_{below1..4}$ . The computation involved depends on the application at hand.

### 2.3 Computing Inter-grid Message

Since the match at the coarser scale should constrain the matching vector at the adjacent finer level, whereas the match at the fine scale should constrain the matching vector at the adjacent coarser level as well, we use this information to couple the 2 systems. The grid lattice in the coarse level is associated with the grid at the fine level. Thus, a value at pixel  $(x^{coarse}, y^{coarse})$  at the coarser level is coupled to the values at pixel  $(x^{fine}, y^{fine})$  where  $Sx^{coarse} \leq x^{fine} < S(x^{coarse} + 1)$  and  $Sy^{coarse} \leq y^{fine} < S(y^{coarse} + 1)$  in the finer level. For example, if the coarser image is 2 times smaller, pixel  $(x, y)$  at the coarse level is coupled to 4 pixels  $(x^{fine}, y^{fine})$  where  $2x^{coarse} \leq x^{fine} < 2(x^{coarse} + 1)$  and  $2y^{coarse} \leq y^{fine} < 2(y^{coarse} + 1)$  at the fine level grid, which are pixels  $(2x, 2y), (2x + 1, 2y), (2x, 2y + 1)$ , and  $(2x + 1, 2y + 1)$ .

Also, the labeling of the coarse level and the fine level should be consistent. This is a novel method of coupling the coarser and finer grids in order to arrive at a solution simultaneously. The inter-grid matching measures if each label map of the finer level agrees with that of the coarser level at the corresponding points of the 2 resolutions. For the scale factor  $S$ , the labeling  $f^{coarse}$  at a coarse level and the labeling  $f^{fine}$  at the corresponding pixels on its contiguous fine level should be consistent, where  $S(f^{coarse} - 1) \leq f^{fine} < S(f^{coarse} + 1)$ .

The inter-grid matching is used to ensure that the solutions on the coarse and fine levels remain consistent. Since the numerical method discretely respects the conservation laws, it is necessary that the amount of the conserved quantities contained in a fine grid be the same as that in the underlying coarse grid region.

$E_{ij}^{INTERGRID(l)}(f_i^l, f_j^{l+1})$  is the inter-grid energy of node  $i$  in level  $l$  and node  $j$  in level  $l + 1$  computed by:

$$E_{ij}^{INTERGRID(l)}(f_i^l, f_j^{l+1}) = g(f_i^l, f_j^{l+1}) \tag{3}$$

where

$$g(x, y) = \begin{cases} 0 & , \text{if } |xS - y| < S - 1 \\ 1 & , \text{otherwise} \end{cases} \tag{4}$$

The inter-grid message is calculated by:

$$m_{pq}^n(f_q^n) \leftarrow \max_{f_p^n} \ell^{-\lambda_I E_{pq}^{INTERGRID(n)}(f_p^n, f_q^{n+1})} m_p^n(f_p^n) \prod_{r \in N(p) \setminus q} m_{rp}^n(f_p^n) \tag{5}$$

where  $p = (x, y)$  on level  $n$ ,  $m_{pq}$  is  $m_{above}$ , when  $q = (x/2, y/2)$  on level  $n + 1$ , or

$$m_{pq}^n(f_q^n) \leftarrow \max_{f_p^n} \ell^{-\lambda_I E_{pq}^{INTERGRID(n)}(f_p^{n-1}, f_q^n)} m_p^n(f_p^n) \prod_{r \in N(p) \setminus q} m_{rp}^n(f_p^n) \tag{6}$$

where  $p = (x, y)$  on level  $n$ ,  $m_{pq}$  is  $m_{below1}, m_{below2}, m_{below3}, m_{below4}$ , when  $q = (2x, 2y), (2x + 1, 2y), (2x, 2y + 1),$  and  $(2x + 1, 2y + 1)$  on level  $n - 1$ .

### 2.4 Our Coarse-and-Fine Belief Propagation Algorithm

The coarse-and-fine belief propagation algorithm which is used to find a solution for an image can be described as:

1. For every resolution level  $n$ , compute the data cost for every pixel  $p$  by:

$$m_p^n(f_p^n) = \ell^{-\lambda_D E_p^{DATA(n)}(f_p^n)}$$

2. All messages  $m_{ij}$  are initialized to a uniform distribution.

3. For each iteration  $t = 1..T$ :

- 3.1 Messages from each pixel  $p$  to each of its neighbor  $q$  within same grid:

$$m_{pq}^n(f_q^n) \leftarrow \max_{f_p^n} \ell^{-\lambda_S E_{pq}^{SMOOTH(n)}(d_p^n, d_q^n)} \prod_{r \in N(p) \setminus q} m_{rp}^n(f_p^n)$$

- 3.2 Messages from each pixel  $p$  to each neighbor  $q$  in the upper grid level:

$$m_{pq}^n(f_q^n) \leftarrow \max_{f_p^n} \ell^{-\lambda_I E_{pq}^{INTERGRID(n)}(f_p^n, f_q^{n+1})} m_p^n(f_p^n) \prod_{r \in N(p) \setminus q} m_{rp}^n(f_p^n)$$

- 3.3 Messages from each pixel  $p$  to each neighbor  $q$  in the lower grid level:

$$m_{pq}^n(f_q^n) \leftarrow \max_{f_p^n} \ell^{-\lambda_I} E_{pq}^{INTERGRID(n)}(f_p^{n-1}, f_q^n) m_p^n(f_p^n) \prod_{r \in N(p) \setminus q} m_{rp}^n(f_p^n)$$

4. After last iteration, compute the belief for each pixel  $p$  at each level:

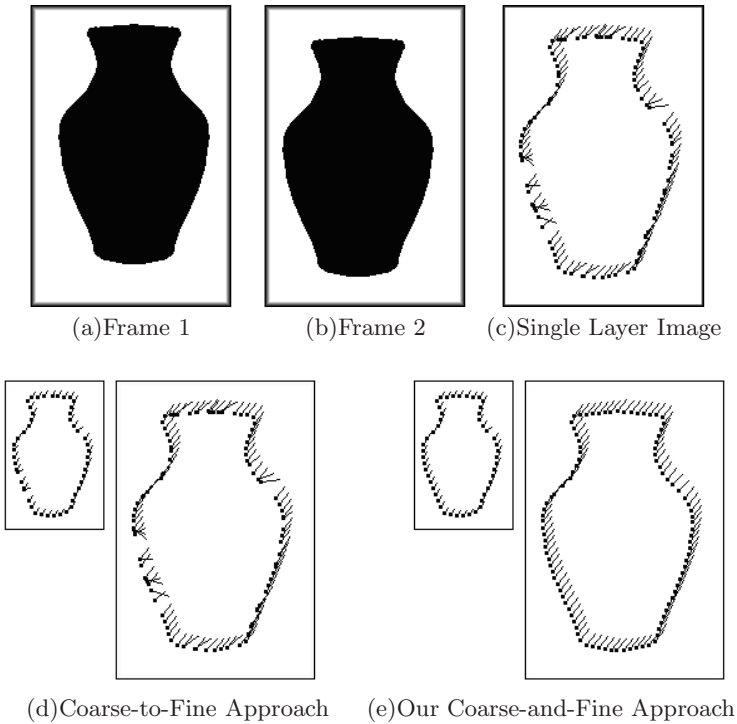
$$b_p^n(f_p^n) \leftarrow m_p^n(f_p^n) \prod_{r \in N(p)} m_{rp}^n(f_p^n)$$

5. Choose the value with maximum belief for each pixel:

$$f_p^n \leftarrow \arg \max_{f_p^n} b_p^n(f_p^n)$$

### 3 Experimental Results

The optical flow for the black jar images are used to compare the results of our coarse-and-fine belief propagation to the one-level and to the classical coarse-to-fine approaches. The single-layer results in Fig. 3c gives poor results compared to the multi-level architectures because the quick convergence of BBP in iteration 1 settles at a poor, local minimum solution. Compared to the single resolution solution, the coarse-to-fine solution in Fig. 3d gives a better result as some distant features are brought closer to each other. However, the algorithm converges early in the coarse level to an incorrect solution at some areas in the contour. This error in the coarse scale solution is incorrectly propagated to the finer scale solution,



**Fig. 3.** Comparing the optical flow of 2 input frames in (a) and (b) using (c) single layer, (d) coarse-to-fine approach, and (e) our coarse to fine approach

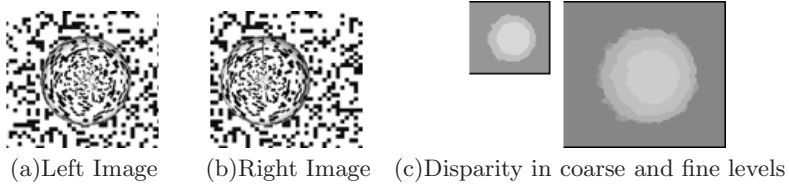


Fig. 4. Stereo result for the synthetic sphere image pair

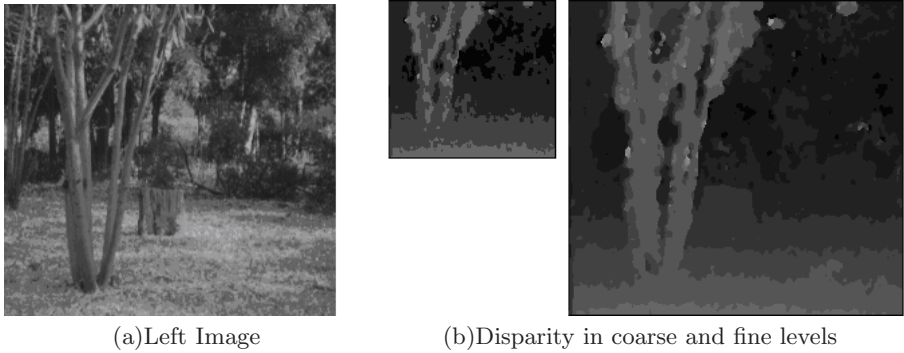
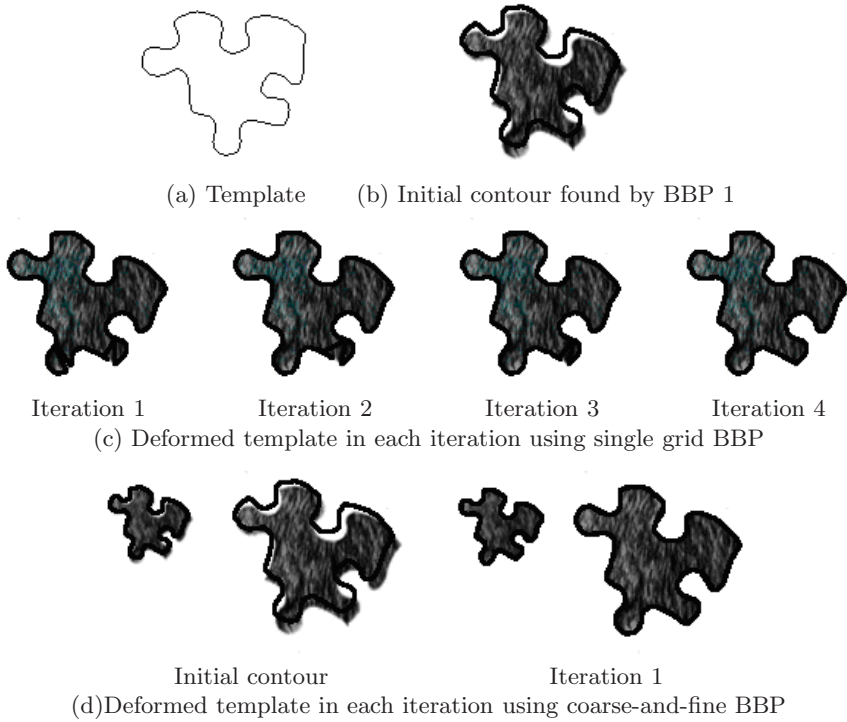


Fig. 5. Stereo result for the real trees image pair

resulting in poor overall performance. Using our coarse-and-fine Bayesian belief propagation method provides perfect results as shown in Fig. 3e.

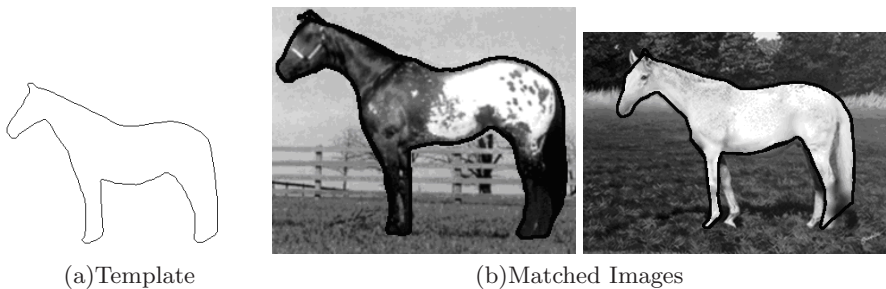
The results of the coarse-and-fine stereo matching for synthetic images and real images are shown in Fig. 4 and 5, respectively. In Fig. 4c, the disparity images are shown in both coarse and fine levels, which appear to be so clean, showing the right shape of the sphere. Fig. 5 shows the result for the tree images which looks very promising.

For the shape matching problem, we use BPP 2 times. First a belief propagation algorithm for contour search [18] is used to search for potential targets by finding vectors  $[(x_c, y_c), S, \beta]$  with possible matches for the position of a shape reference point (an arbitrary center of a contour shape), scale  $S$ , and rotation  $\beta$ . Then, the template is transformed and placed in the target image using each potential transformation vector  $[(x_c, y_c), S, \beta]$  and a second deformable template matching using another coarse-and-fine Bayesian belief propagation snake algorithm for contour matching algorithm is performed. This contour matching is very efficient and converges relatively quickly even for a large search space that we used with  $(\Delta x, \Delta y) = (u, v)$  matched values each ranging between -20 to 20. Bayesian belief propagation is known to be a slow method due to its computational complexity. Nevertheless, it gives more reliable results and converges within very few iterations when compared to other optimization algorithms. For example, the Gibbs sampler with simulated annealing we used in [3] converges after about 300 iterations. In most cases, the BBP process con-verges in less than 5 iterations. Single grid layer matching also works well for the contour matching



**Fig. 6.** A comparison of contour matching between single resolution BBP and coarse-and-fine BBP

because the template is allowed to be flexibly deformed. However, the coarse-and-fine BBP matching converges in a fewer iterations as the matched points are closer in the coarse level while the detailed information is provided by the fine level. Fig. 6 shows a comparison between the deformed contour in one resolution grid and the coarse-and-fine multi-resolution approaches including results for a few intermediate iterations during the matching process of a small jig-saw



**Fig. 7.** Matching a horse shape template using coarse-and-fine BPP

puzzle shape. The single grid BBP moves close to the optimum solution after the first iteration, but it still has some error, requiring a few more iterations to yield a correct match. In comparison, the coarse-and-fine algorithm achieves a correct deformation of the template to match the target in a single iteration. For this instance, the single level completed the processing in 30 seconds while the coarse-and-fine method worked in 12 seconds. The Gibbs sampler coarse-and-fine approach requires approximate 75 seconds to converge. Fig. 7 illustrates contour matching for a horse shape template, showing the results of matching the shape in two different horse images.

## 4 Conclusion

We propose the use of a coarse-and-fine Bayesian belief propagation (BBP) optimization method to solve various computer vision modules that involve the correspondence problem including shape matching, stereovision, and optical flow. The algorithm works with a grid matching, and an inter-grid correspondence model in terms of message passing in a Bayesian belief propagation (BBP) network. The local smoothness and other constraints are expressed within each resolution scale grid and also between grids in a single paradigm. The coarse-and-fine algorithm uses matching results in each layer to constrain the process in its adjacent layers by measuring the consistency between corresponding points between itself and those in both the higher and lower adjacent layers so that good matches at multiple levels constrain one another. The method is used to constrain a solution with examples given in deformable templates, stereovision, and optical flow to demonstrate the desirable properties our the BPP algorithm such as rapid convergence, the ability to obtain near optimal solution while avoiding local minima, and immunity to error propagation found in the coarse-to-fine approach.

The experiments show that the coarse level started to converge earlier as it brings distant features closer, but it still used the details in the fine level to refine itself in the right direction towards an optimal correspondence solution. Meanwhile, the fine level receives messages from the coarser level which carries some distant features so that it can escape from the local minimum to obtain a correct solution.

## References

1. Horn, K.B.P, Schunck, B.: Determining Optical Flow. *Artificial Intelligence* 17, 185–203 (1981)
2. Besag, J.: On the Statistical Analysis of Dirty Pictures. *Journal of Royal Statistical Society Series B. Methodological* 48(3), 259–302 (1986)
3. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6(6), 721–741 (1984)
4. Bilbro, G.L., Snyder, W.E: Optimization of Functions with Many Minima. *IEEE Transactions on Systems* 21(4), 840–849 (1991)



5. Amini, A., Weymouth, T., Jain, R.: Using Dynamic Programming for Solving Variational Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(9), 855–867 (1990)
6. Boykov, Y., Veksler, O., Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(11), 1222–1239 (2001)
7. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*, pp. 77–286. Morgan Kaufmann Publisher, California (1988)
8. Weiss, Y.: Belief Propagation and Revision in Networks with Loops. MIT AI Memo. Vol. AIM-, No. CBCL-155, 1616 (1997) 1-14
9. Weiss, Y.: Interpreting Images by Propagating Bayesian Beliefs. *Advance in Neural Information Processing Systems* 9(1), 908–915 (1997)
10. Shental, N., Zomet, A., Hertz, T., Weiss, Y.: Learning and Inferring Image Segmentations Using the GBP Typical Cuts Algorithm. In: *Proceedings of IEEE International Conference on Computer Vision*, vol. 2, pp. 1243–1250. IEEE Computer Society Press, Los Alamitos (2003)
11. Komogorov, V.: Convergent Tree-reweighed Message Passing for Energy Minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(10), 1568–1583 (2006)
12. Sun, J., Shum, H.-Y., Zheng, N.-N.: Stereo Matching Using Belief Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(7), 1–14 (2003)
13. Coughlan, J.M., Ferreira, S.J.: Finding Deformable Shapes Using Loopy Belief Propagation. In: *Proceedings of European Conference on Computer Vision-Part III*, pp. 453–468 (2003)
14. Gao, J., Shi, J.: Multiple frame motion inference using belief propagatio. In: *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 875–880. IEEE Computer Society Press, Los Alamitos (2004)
15. Marapane, S.B., Trivedi, M.M.: Multi-Primitive Hierarchical (MPH) Stereo Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(3), 227–240 (1994)
16. Jain, A.K., Zhong, Y., Lakshmanan, S.: Object Matching Using Deformable Templates. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 18(3), 267–278 (1996)
17. Felzenwalb, P., HuttenLocker, D.: Efficient Belief Propagation for Early Visio. *IEEE Conference of Computer Vision and Pattern Recognition* 1, 1261–1268 (2004)
18. Tipwai, P.: Ph.D. Dissertation. King Monkut’s University of Technology Thonburi (2007)

# 3D Object Recognition Based on Low Frequency Response and Random Feature Selection

Roberto A. Vázquez, Humberto Sossa, and Beatriz A. Garro

Centro de Investigación en Computación – IPN  
Av. Juan de Dios Batiz, esquina con Miguel de Othon de Mendizábal  
Ciudad de México, 07738, México  
ravem@ipn.mx, hsossa@cic.ipn.mx, bgarrol@ipn.mx

**Abstract.** In this paper we propose a view-based method for 3D object recognition based on some biological aspects of infant vision. The biological hypotheses of this method are based on the role of the response to low frequencies at early stages, and some conjectures concerning how an infant detects subtle features (stimulating points) from an object. In order to recognize an object from different images of it (different orientations from  $0^\circ$  to  $100^\circ$ ) we make use of a dynamic associative memory (DAM). As the infant vision responds to low frequencies of the signal, a low-filter is first used to remove high frequency components from the image. Then we detect subtle features in the image by means of a random feature selection detector. At last, the DAM is fed with this information for training and recognition. To test the accuracy of the proposal we use the Columbia Object Image Library (COIL 100) database.

## 1 Introduction

View-based object recognition has attracted much attention in recent years. In contrast to methods that rely on pre-defined geometric (shape) models for recognition, view-based methods learn a model of the object's appearance in a two-dimensional image under different poses and illumination conditions.

Several view-based methods have been proposed to recognize 3D objects. In Poggio and Edelman [2] show that 3D objects can be recognized from the raw intensity values in 2D images (*pixel-based representation*) using a network of generalized radial basis functions. Turk and Pentland [3] demonstrate that human faces can be represented and recognized by eigenfaces. Representing a face image as a vector of pixel values, the eigenfaces are the eigenvectors associated with the largest eigenvalues that are computed from a covariance matrix of the sample vectors. An attractive feature of this method is that the eigenfaces can be learned from the sample images in pixel representation without any feature selection. Despite this method is a computationally expensive technique; it has been used in different vision tasks from face recognition to object tracking. Murase and Nayar [4] and [5] developed a parametric eigenspace method to recognize 3D objects directly from their appearance. For each object of interest, a set of images in which the object appears in different poses is obtained as training examples. Next, the eigenvectors are computed from the covariance matrix of the training set. The set of images is projected to a low dimensional subspace spanned

by a subset of eigenvectors, in which the object is represented as a manifold. A compact parametric model is constructed by interpolating the points in the subspace. In recognition, the image of a test object is projected to the subspace and the object is recognized based on the manifold it lies on.

General-purpose learning methods such as support vector machines (SVMs) have also been used for this problem. Schölkopf [7] was the first to apply SVMs to recognize 3D objects from 2D images and has demonstrated the potential of this approach in visual learning. Pontil and Verri [8] also used SVMs for 3D object recognition and experimented with a subset of the COIL-100 data set. Their training set consisted of 36 images (one for every  $10^\circ$ ) for each of the 32 objects they chose, and the test sets consist of the remaining 36 images for each object. For 20 random selections of 32 objects from the COIL-100, the system achieves perfect recognition rate. A subset of COIL-100 set has also been used by Roobaert and Van Hulle [9] to compare the performance of SVMs with different pixel-based input representations.

In this research, we propose a view-based method for 3D object recognition based on some biological aspects of infant vision. The biological hypotheses of this proposal are based on the role of the response to low frequencies at early stages, and some conjectures concerning how an infant detects subtle features (stimulating points) in a face or object [10], [11], [14] and [15]. As a learning device we use a dynamic associative memory (DAM) used to recognize different images of objects at different orientations (from  $0^\circ$  to  $90^\circ$ ). Due to the infant vision responds to low frequencies of the signal, a low-filter is first used to remove high frequency components from the image. Then we detect subtle features in the image by means of a random selection of stimulating points. At last, the DAM is fed with this information for training and recognition. To test the accuracy of the proposal, we use the Columbia Object Image Library (COIL 100) [6]. The training set consists of 100 images (one for every object at  $0^\circ$ ), and the testing set consists of the 20 images (from  $5$  to  $100^\circ$ ) for each object.

## 2 Dynamic Associative Memory

The Dynamic associative model is not an iterative model as Hopfield's model. This model emerges as an improvement of the model proposed in [13] and some of the results presented in [16].

Let  $\mathbf{x} \in \mathbf{R}^n$  and  $\mathbf{y} \in \mathbf{R}^m$  an input and output pattern, respectively. An association between input pattern  $\mathbf{x}$  and output pattern  $\mathbf{y}$  is denoted as  $(\mathbf{x}^k, \mathbf{y}^k)$ , where  $k$  is the corresponding association. Associative memory  $\mathbf{W}$  is represented by a matrix whose components  $w_{ij}$  can be seen as the synapses of the neural network. If  $\mathbf{x}^k = \mathbf{y}^k \forall k = 1, \dots, p$  then  $\mathbf{W}$  is auto-associative, otherwise it is hetero-associative. A distorted version of a pattern  $\mathbf{x}$  to be recalled will be denoted as  $\tilde{\mathbf{x}}$ . If an associative memory  $\mathbf{W}$  is fed with a distorted version of  $\mathbf{x}^k$  and the output obtained is exactly  $\mathbf{y}^k$ , we say that recalling is robust.

### 2.1 Building the Associative Memory

Due to several regions of the brain interact together in the process of learning and recognition [12], in the dynamic model there are defined several interacting areas; also it integrated the capability to adjust synapses in response to an input stimulus. Before the brain processes an input pattern, it is hypothesized that it is transformed and codified by the brain. This process is simulated using the procedure introduced in [13].

This procedure allows computing *codified patterns* from input and output patterns denoted by  $\bar{\mathbf{x}}$  and  $\bar{\mathbf{y}}$  respectively;  $\hat{\mathbf{x}}$  and  $\hat{\mathbf{y}}$  are *de-codifying patterns*. Codified and de-codifying patterns are allocated in different interacting areas and  $d$  defines of much these areas are separated. On the other hand,  $d$  determines the noise supported by our model. In addition a simplified version of  $\mathbf{x}^k$  denoted by  $s_k$  is obtained as:

$$s_k = s(\mathbf{x}^k) = \mathbf{mid} \mathbf{x}^k \tag{1}$$

where **mid** operator is defined as  $\mathbf{mid} \mathbf{x} = x_{(n+1)/2}$ .

When the brain is stimulated by an input pattern, some regions of the brain (interacting areas) are stimulated and synapses belonging to those regions are modified. In this model, the most excited interacting area is call *active region* (AR) and could be estimated as follows:

$$ar = r(\mathbf{x}) = \arg \left( \min_{i=1}^p |s(\mathbf{x}) - s_i| \right) \tag{2}$$

Once computed the *codified patterns*, the *de-codifying patterns* and  $s_k$  we can build the associative memory.

Let  $\{(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k) | k = 1, \dots, p\}$ ,  $\bar{\mathbf{x}}^k \in \mathbf{R}^n$ ,  $\bar{\mathbf{y}}^k \in \mathbf{R}^m$  a fundamental set of associations (codified patterns). Synapses of associative memory  $\mathbf{W}$  are defined as:

$$w_{ij} = \bar{y}_i - \bar{x}_j \tag{3}$$

After computed the *codified patterns*, the *de-codifying patterns*, the reader can easily corroborate that any association can be used to compute the synapses of  $\mathbf{W}$  without modifying the results. In short, building of the associative memory can be performed in three stages as:

1. Transform the fundamental set of association into codified and de-codifying patterns by means of previously described Procedure 1.
2. Compute simplified versions of input patterns by using equation 1.
3. Build  $\mathbf{W}$  in terms of codified patterns by using equation 3.

### 2.2 Modifying Synapses of the Associative Model

There are synapses that can be drastically modified and they do not alter the behavior of the associative memory. In the contrary, there are synapses that only can be slightly modified to do not alter the behavior of the associative memory; we call this set of synapses *the kernel* of the associative memory and it is denoted by  $\mathbf{K}_w$ .

Let  $\mathbf{K}_w \in \mathbf{R}^n$  the kernel of an associative memory  $\mathbf{W}$ . A component of vector  $\mathbf{K}_w$  is defined as:

$$kw_i = \mathbf{mid}(w_{ij}), j = 1, \dots, m \quad (4)$$

According to the original idea of our proposal, synapses that belong to  $\mathbf{K}_w$  are modified as a response to an input stimulus. Input patterns stimulate some *active regions*, interact with these regions and then, according to those interactions, the corresponding synapses are modified. Synapses belonging to  $\mathbf{K}_w$  are modified according to the stimulus generated by the input pattern. This adjusting factor is denoted by  $\Delta w$  and can be computed as:

$$\Delta w = \Delta(\mathbf{x}) = s(\bar{\mathbf{x}}^r) - s(\mathbf{x}) \quad (5)$$

where  $r$  is the index of the *active region*.

Finally, synapses belonging to  $\mathbf{K}_w$  are modified as:

$$\mathbf{K}_w = \mathbf{K}_w \oplus (\Delta w_{new} - \Delta w_{old}) \quad (6)$$

where operator  $\oplus$  is defined as  $\mathbf{x} \oplus e = x_i + e \forall i = 1, \dots, m$ . As you can appreciate, modification of  $\mathbf{K}_w$  in equation 6 depends of the previous value of  $\Delta w$  denoted by  $\Delta w_{old}$  obtained with the previous input pattern. Once trained the **AM**, when it is used by first time, the value of  $\Delta w_{old}$  is set to zero.

### 2.3 Recalling a Pattern Using the Proposed Model

Once synapses of the associative memory have been modified in response to an input pattern, every component of vector  $\bar{\mathbf{y}}$  can be recalled by using its corresponding input vector  $\bar{\mathbf{x}}$  as:

$$\bar{y}_i = \mathbf{mid}(w_{ij} + \bar{x}_j), j = 1, \dots, n \quad (7)$$

In short, pattern  $\bar{\mathbf{y}}$  can be recalled by using its corresponding key vector  $\bar{\mathbf{x}}$  or  $\tilde{\mathbf{x}}$  in six stages as follows:

1. Obtain index of the active region  $ar$  by using equation 2.
2. Transform  $\mathbf{x}^k$  using de-codifying pattern  $\hat{\mathbf{x}}^{ar}$  by applying the following transformation:  $\tilde{\mathbf{x}}^k = \mathbf{x}^k + \hat{\mathbf{x}}^{ar}$ .
3. Compute adjust factor  $\Delta w = \Delta(\tilde{\mathbf{x}})$  by using equation 5.
4. Modify synapses of associative memory  $\mathbf{W}$  that belong to  $\mathbf{K}_w$  by using equation 6.
5. Recall pattern  $\hat{\mathbf{y}}^k$  by using equation 7.
6. Obtain  $\mathbf{y}^k$  by transforming  $\hat{\mathbf{y}}^k$  using de-codifying pattern  $\hat{\mathbf{y}}^{ar}$  by applying transformation:  $\mathbf{y}^k = \hat{\mathbf{y}}^k - \hat{\mathbf{y}}^{ar}$ .

The formal set of prepositions that support the correct functioning of this dynamic model and the main advantages with respect to other classical models can be found in [21]. Some interesting applications of this model are described in [17], [18], [19], and [20].

### 3 Description of the Proposal

The proposal consists of a DAM used to recognize different images of a 3D object. As the infant vision responds to low frequencies of the signal, a low-filter is first used to remove high frequency components from the image. Then we detect subtle features in the image by means of a random selection of stimulating points. At last, the DAM is fed with this information for training and recognition. In Fig. 1 it is shown a general schema of the proposal.

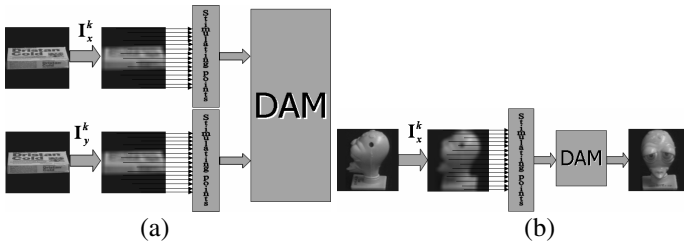


Fig. 1. A general schema of the proposal. (a) Building phase. (b) Recall phase.

#### 3.1 Response to Low Frequencies

It is important to mention that instead of using a filter that exactly simulates the infant vision system behavior at any stage we use a low-pass filter to remove high frequency. This kind of filter could be seen as a slight approximation of the infant vision system due to it eliminates high frequency components from the pattern.

For simplicity, we used an average filter. If we apply this filter to an image, the resultant image could be hypothetically seen as the image that infants perceive in a specific stage of their life.

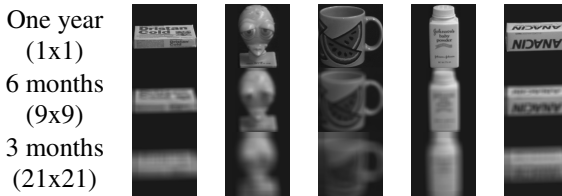


Fig. 2. Images filtered with masks of different size. Each group could be associated with different stages of infant vision system.

For example, we could associate the size of the mask used in the filter with a specific stage. For example if the size of the mask is one we could say that the resultant

image corresponds to one year of age; if the size of the mask is the biggest we could say that the resultant image corresponds to a newborn. Fig. 2 shows some images filtered with masks of different sizes.

### 3.2 Random Selection

In order to simulate the random selection of the infant vision system we add to the DAM model a vector of stimulating points **SP** where each stimulating point, given by  $sp_i = random(n)$ , is a random number between zero and the length of input pattern and  $i = 1, \dots, c$  where  $c$  is the number of stimulating points used. To determine the active region we allocate in the DAM model an alternative simplified version of each pattern  $\mathbf{x}^k$  given by:

$$ss_i^k = ss(\mathbf{x}^k) = \mathbf{x}_{sp_i}^k \tag{8}$$

Once compute these simplified versions we could estimate the active region as follows:

$$ar = r(\mathbf{x}) = \arg \max_{i=1}^p(\mathbf{a}) \tag{9}$$

where  $a_{b_i} = a_{b_i} + 1$ ,  $b_i = \arg \min_{k=1}^p \left| [ss(\mathbf{x})]_i - ss_i^k \right|$  and  $i = 1, \dots, c$ .

We supposed that most relevant information that best describes an object in an image is concentrated in the center of the image. In general, when humans pay attention to a particular object, most of the time humans they focus their sight in the center of the field vision. Trying to simulate this, we use a Gaussian random number generator based in the polar form of the Box-Muller transformation [1].

### 3.3 Implementation of the Proposal

Building of the DAM is done as follows:

Let  $\mathbf{I}_x^k$  and  $\mathbf{I}_y^k$  an association of images and  $C$  be the number of stimulating points.

1. Take at random a stimulating point  $sp_i, i = 1, \dots, c$ .
2. For each association:
  - a. Select filter size and apply it to the stimulating points in the images.
  - b. Transform the images into a vector  $(\mathbf{x}^k, \mathbf{y}^k)$  by means of the standard image scan method.
3. Train the DAM as in building procedure and compute the alternative simplified version of the patterns by using equation 8.

Pattern  $\mathbf{I}_y^k$  can be recalled by using its corresponding key image  $\mathbf{I}_x^k$  or distorted version  $\tilde{\mathbf{I}}_x^k$  as follows:

1. Use the same stimulating point,  $sp_i, i=1, \dots, c$  and filter size as in building phase.
2. Apply filter to the stimulating points in the images.
3. Transform the images into a vector by means of the standard image scan method
4. Determine active region using equation 9.
5. Apply steps from two to six as described in recalling procedure.

## 4 Experimental Results

To test the accuracy of the proposal, we have used the Columbia Object Image Library (COIL 100). The training set consists of 100 images (one for every object at  $0^\circ$ ), and the testing set consists of the 20 images (from  $5$  to  $100^\circ$ ) for each object. Each photo is in colour and of  $128 \times 128$  pixels.

The DAM was trained in the auto-associative way using building procedure described in section 3.3. Once trained the DAM we proceeded to test the proposal with three sets of experiments. In the first set of experiments we show how by using a Gaussian number generator and different stimulating points the accuracy of the proposal increases. In the second set of experiment we show how by changing the standard deviation and different stimulating points the accuracy of the proposal increases. Finally in the third set experiments, we show how by increasing the size of the filter the accuracy of the proposal could be also increased.

### 4.1 First Set of Experiments

In this set of experiments we compared four random number generators. The first generator generates uniformly distributed random numbers (uniform 1) in intervals. For the details, refer to [20]. The second generator also generates uniformly distributed random numbers (uniform 2). The third and forth ones generate Gaussian random numbers based on the polar form of the Box-Muller transformation. For the third generator, we generated random numbers over the image transformed into a vector using a mean of 8191.5 and a standard deviation of 4729.6. For the forth generator we generated random numbers over axis  $x$  and  $y$  of the image by using a mean of 63.5 and a standard deviation of 37.09. By using this generator we tried to approximate the way humans focus their sight to the center of the field vision. We have also experimented with different numbers of stimulating points.

As you can appreciate from Fig. 3(a), in average the accuracy of the proposal when using the first two generators is of 36% and 34% respectively. In general, the accuracy of the proposal when using the Gaussian generator increases; in average for Gaussian 1 and Gaussian 2 the accuracy of the proposal is 42% and 50% respectively. On the other hand, when augmenting to 200 stimulating points the accuracy of the proposal tends to increase.

Despite of the accuracy obtained using each generator, we can see a clearly advantage of generate Gaussian random numbers over axis  $x$  and  $y$  on an image against the other generators.



## 4.2 Second Set of Experiments

In this set of experiments we modified the value of the standard deviation of forth generator in order to improve the accuracy of the proposal. We used this generator in order to approximate the way as a human focus their sight at the center of the field vision. Despite humans focus their sight at the center of all field vision, they also perceive information from the periphery to the center field vision. We could control the radius of the center to the periphery by adjusting the value of standard deviation.

In the previous set of experiments we used a standard deviation of 37.09, this means that we generated stimulating points from the whole image. If we reduce the standard deviation we could concentrate the stimulating points to the center of the image.

In Fig. 3(b) we show the accuracy of the proposal when varying the value of standard deviation. SD-x is the value of the standard deviation 37.09 minus the value of x. As you can be appreciated from this figure, if we reduce the standard deviation the accuracy of the proposal increases, but when the stimulating points are too concentrated to the center of the image the accuracy of the proposal tends to decrease.

In average the accuracy of the proposal for the different standard deviations SD-5, SD-10, SD-15, SD-20, SD-25 and SD-30 was of 56%, 72%, 83%, 87%, 76% and 62% respectively.

It is worth mentioning from this experiment how the proposal's accuracy can be increased when changing the value of the standard deviation. Particularly, the best accuracy was of 92% when using 2001 stimulation points and SD-20.

## 4.3 Third Set of Experiments

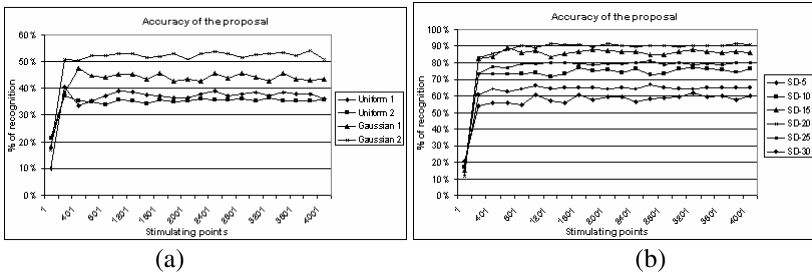
In this set of experiments we removed the high frequencies of the images in order to improve the accuracy of the proposal. By removing high frequencies, as we will see, contributes to eliminate unnecessary information and help the DAM to learn efficiently objects. As we previously said, we have used an average filter. In this set of experiments we applied the filter to the stimulating point in the images. We tested different size of the filter from 1 to 39 combine with different SD-x.

As you can appreciate from Fig. 4(a), the accuracy of the proposal when using SD-0 increases when the size of the filter is increased. By using 1000, 2000 and 3000 stimulating points we reached an accuracy of 97%, 95% and 96% respectively. After a filter of size 35 the accuracy starts to decrease.

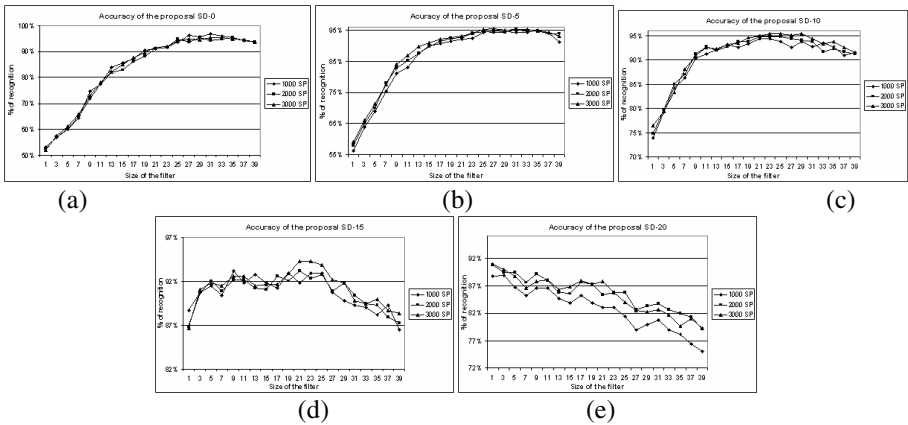
The accuracy of the proposal using SD-5 increases when the size of the filter is increased, see Fig. 4(b). By using 1000, 2000 and 3000 stimulating points we reached an accuracy of 95%, 95% and 96% respectively. Also, after a filter of size 35 the accuracy starts to decrease.

In Fig. 4(c) it is shown how the accuracy of the proposal when using SD-15 increases when the size of the filter is increased. By using 1000, 2000 and 3000 stimulating points we reached an accuracy of 94%, 95% and 96% respectively. After a filter of size 29 the accuracy starts to decrease.

As in the case of previous configurations, the accuracy of the proposal when using SD-20 increases when the size of the filter is increased as shown in Fig. 4(d). By



**Fig. 3.** (a) Accuracy of the proposal using different random number generators. (b) Accuracy of the proposal using different standard deviation values.



**Fig. 4.** (a-e) Accuracy of the proposal using different size of the filter and SD-x

using 1000, 2000 and 3000 stimulating points we reached an accuracy of 93%, 93% and 94% respectively. After a filter of size 21 the accuracy starts to decrease.

In the contrary, as you can appreciate from Fig. 4(e), the accuracy of the proposal when using SD-25 decreases when the size of the filter is increased. By using 1000, 2000 and 3000 stimulating points we reached an accuracy of 89%, 91% and 91% respectively.

Through several experiments we have observed that after applying a filter of size greater than 1 and SD-25 the accuracy of the proposal tends to diminish. In average, by removing high frequencies and by selecting at random stimulating points, and by using a Gaussian number generator over axis  $x$  and  $y$ , contributes to eliminating unnecessary information and help the DAM to learn efficiently the objects. In general, the accuracy of the proposal surpasses the 90% of recognition and with some configurations we up-performed this result. By using SD-0, 1000 stimulating points and a filter of size 31 we reached an accuracy of 97%.

The results obtained with the proposal through several experiments were comparable with those obtained by means of a PCA-based method (99%). Although PCA is a powerful technique it consumes a lot of time to reduce the dimensionality of the data.

Our proposal, because of its simplicity in operations, is not a computationally expensive technique and the results obtained are comparable to those provided by PCA.

## 5 Conclusions

In this paper we have proposed a view-based method 3D object recognition based on some biological aspects of infant vision. We have shown that by applying some aspects of the infant vision system it is possible to enhance the performance of an associative memory and also make possible its application to complex problems such as 3D object recognition.

The biological hypotheses of this method are based on the role of the response to low frequencies at early stages, and some conjectures concerning to how an infant detects subtle features (stimulating points) in objects.

We used a DAM used to recognize different images of a 3D object. As the infant vision responds to low frequencies of the signal, a low-filter is first used to remove high frequency components from the image. Then we detect subtle features in the image by means of a random selection of stimulating points. At last, the DAM is fed with this information for training and recognition.

Through several experiments we have shown how the accuracy of the proposal can be increased by using a Gaussian number generator over axis  $x$  and  $y$  on an image. Trying to approximate the way as a human focus their sight to the center of the field vision and perceive information from the periphery to the center field vision. We could control the radius of the center to the periphery adjusting the value of standard deviation.

By removing high frequencies and by randomly selecting of stimulating points contributes to eliminate unnecessary information and help the DAM to learn efficiently the objects. In general, the accuracy of the proposal oscillates between 90% and 97%. Important to mention is that, to our knowledge, nobody has reported results of this type using an associative memory for 3D object recognition.

The results obtained with the proposal were comparable with those obtained by means of a PCA-based method. Although PCA is a powerful technique it consumes a lot of time to reduce the dimensionality of the data. Our proposal, because of its simplicity in operations, is not a computationally expensive technique and the results obtained are comparable to those provided by PCA.

**Acknowledgments.** This work was economically supported by SIP-IPN under grant 20071438 and CONACYT under grant 46805.

## References

- [1] Box, G.E.P., Muller, M.E.: A note on the generation of random normal deviates. *The Annals of Mathematical Statistics* 29(2), 610–611 (1958)
- [2] Poggio, T., Edelman, S.: A network that learns to recognize 3d objects. *Nature* 343, 263–266 (1990)
- [3] Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3(1), 71–86 (1991)

- [4] Murase, H., Nayar, S.K.: Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision* 14, 5–24 (1995)
- [5] Nayar, S.K., Nene, S.A., Murase, H.: Real-time 100 object recognition system. In: *Proceedings of IEEE International Conf. on Robotics and Automation*, pp. 2321–2325. IEEE Computer Society Press, Los Alamitos (1996)
- [6] Nayar, S.K., Nene, S.A., Murase, H.: Columbia Object Image Library (COIL 100). Tech. Report No. CUCS-006-96. Department of Comp. Science, Columbia University
- [7] Schölkopf, B.: Support Vector Learning. PhD thesis, Informatik der Technischen Universität Berlin (1997)
- [8] Pontil, M., Verri, A.: Support vector machines for 3d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(6), 637–646 (1998)
- [9] Roobaert, D., Hulle, M.V.: View-based 3d object recognition with support vector machines. In: *Proceedings of IEEE International Workshop on Neural Networks for Signal Processing*, pp. 77–84. IEEE Computer Society Press, Los Alamitos (1999)
- [10] Mondloch, C.J., et al.: Face Perception During Early Infancy. *Psychological Science* 10(5), 419–422 (1999)
- [11] [11] Acerra, F., Burmod, Y., Schonen, S.: Modelling aspects of face processing in early infancy. *Developmental science* 5(1), 98–117 (2002)
- [12] Laughlin, S.B., Sejnowski, T.J.: Communication in neuronal networks. *Science* 301, 1870–1874 (2003)
- [13] Sossa, H., Barrón, R., Vázquez, R.A.: Transforming Fundamental set of Patterns to a Canonical Form to Improve Pattern Recall. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) *IBERAMIA 2004. LNCS (LNAI)*, vol. 3315, pp. 687–696. Springer, Heidelberg (2004)
- [14] Slaughter, V., Stone, V.E., Reed, C.: Perception of Faces and Bodies Similar or Different? *Current Directions in Psychological Science* 13(9), 219–223 (2004)
- [15] Cuevas, K., Rovee-Collier, C., Learmonth, A.E.: Infants Form Associations Between Memory Representations of Stimuli That Are Absent. *Psychological Science* 17(6), 543–549 (2006)
- [16] Sossa, H., Barron, R., Vazquez, R.A.: Study of the Influence of Noise in the Values of a Median Associative Memory. In: Beliczynski, B., et al. (eds.) *ICANNGA 2007, Part II. LNCS*, vol. 4432, pp. 55–62. Springer, Heidelberg (2007)
- [17] Vazquez, R.A., Sossa, H.: Associative Memories Applied to Image Categorization. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) *CIARP 2006. LNCS*, vol. 4225, pp. 549–558. Springer, Heidelberg (2006)
- [18] Vazquez, R.A., Sossa, H., Garro, B.A.: A New Bi-directional Associative Memory. In: Gelbukh, A., Reyes-Garcia, C.A. (eds.) *MICAI 2006. LNCS (LNAI)*, vol. 4293, pp. 367–380. Springer, Heidelberg (2006)
- [19] Vazquez, R.A., Sossa, H.: A computational approach for modeling the infant vision system in object and face recognition. *Journal BMC Neuroscience* 8(suppl 2), P204 (2007)
- [20] Vazquez, R.A., Sossa, H., Garro, B.A.: Low frequency responses and random feature selection applied to face recognition. In: Kamel, M., Campilho, A. (eds.) *ICIAR 2007. LNCS*, vol. 4633, pp. 818–830. Springer, Heidelberg (2007)
- [21] Vazquez, R.A., Sossa, H.: A new associative memory with dynamical synapses to be submitted (2007)

# Image Processing for 3D Reconstruction Using a Modified Fourier Transform Profilometry Method

Jesus Carlos Pedraza Ortega<sup>1</sup>, Jose Wilfrido Rodriguez Moreno<sup>2</sup>,  
Leonardo Barriga Rodriguez<sup>1</sup>, Efren Gorrostieta Hurtado<sup>3</sup>, Tomas Salgado Jimenez<sup>1</sup>,  
Juan Manuel Ramos Arreguin<sup>4</sup>, and Angel Rivas<sup>5</sup>

<sup>1</sup> Centro de Ingenieria y Desarrollo Industrial, Av. Pie de la Cuesta No. 702 Desarrollo San Pablo, Queretaro, Qro. C.P. 76130 Mexico

jpedraza@cidesi.mx, lbarriga@cidesi.mx, tsalgado@cidesi.mx

<sup>2</sup> VALEO - Front End Module Division, Carr. Fed. Ags. a Lagos de Moreno Km. 75, Aguascalientes, Ags. C.P. 20340 Mexico

jose-wilfrido.rodriguez@valeo.com

<sup>3</sup> Universidad Autonoma de Queretaro, Centro Universitario, Cerro de las Campanas Queretaro, Qro. C.P. 76010 México

efren.hurtado@usa.net

<sup>4</sup> Universidad Tecnologica de San Juan del Rio, Av. La Palma 125, Col. Vista Hermosa, San Juan del Rio, Queretaro, Qro. C.P. 76800 México

jsistdig@yahoo.com

<sup>5</sup> Universidad Autonoma de Nuevo Leon - FIME, Av. Universidad s/n, Cd. Universitaria, San Nicolas de los Garza, Nuevo Leon, C.P. 66451 México

rrv@hotmail.com

**Abstract.** An image processing algorithm based on the Fourier Transform Profilometry (FTP) method for 3D reconstruction purposes is presented. This method uses a global and local analysis for the phase unwrapping stage and obtains better results than using a simple unwrapping algorithm in the normal FTP method. A sinusoidal fringe pattern of known spatial frequency is firstly projected on a reference frame and an image is acquired. Then, the object of the shape to know is placed in front of the reference frame, and the same fringe pattern is projected. Once again another image is acquired. The projected pattern is distorted according to the shape of the object. Later, the modified Fourier Transform Profilometry method is applied to the acquired images. The digitized images contains the  $(x,y)$  pixels of the object including the fringe pattern, and the phase difference between the acquired images contains the  $z$  (height or depth) information. The novelty in the proposed method comes in the part of the unwrapping algorithm at the moment of obtaining the depth information by using the above mentioned combined analysis.

**Keywords:** Unwrapping algorithms, local and global analysis, Fourier Transform Profilometry, 3D reconstruction.

## 1 Introduction

In the past 30 years, an important area of research in Computer Vision has been the inference of the 3D information about a scene from its 2D images. The idea is to

extract the useful depth information from an image in an efficient and automatic way. The obtained information can be used to guide various processes such as robotic manipulation, automatic inspection, inverse engineering, 3D depth map for navigation and virtual reality applications [1]. Depending on the application, a simple 3D description is necessary to understand the scene and perform the desired task, while in other cases a dense map or detailed information of the object's shape is necessary. Moreover, in some cases a complete 3D description of the object may be required.

Today, the three-dimensional shape of an object can be obtained in different ways. In 3D machine vision can be classified in two categories; Active and Passive Methods, which can be also classified as contact and non contact methods.

The active methods project energy in the scene and detect the reflected energy; some examples of these methods are sonar, laser ranging, fringe projection and structured method. The active methods work based on triangulation. On other hand, the passive methods use ambient illumination during data acquisition; some examples of these methods are motion parallax, shape from shading, stereo vision, depth from defocus and depth from focus.

In the present work we present a modification algorithm to the active method called the Fourier Transform Profilometry. The contribution is to add a pre-processing filter plus a data analysis in the unwrapping step. The method will be explained in the next section.

## 2 Fourier Transform Profilometry

The image of an object with projected fringes can be represented by the following equation:

$$g(x, y) = a(x, y) + b(x, y) * \cos[2 * \pi f_0 x + \varphi(x, y)] \quad (1)$$

where  $g(x,y)$  is the intensity of the image at  $(x,y)$  point,  $a(x,y)$  represents the background illumination,  $b(x,y)$  is the contrast between the light and dark fringes,  $f_0$  is the spatial-carrier frequency and  $\varphi(x,y)$  is the phase corresponding to the distorted fringe pattern, observed from the camera. The experimental setup will be explained in the next section.

Here is important to say that  $\varphi(x,y)$  contains the desired information, and  $a(x,y)$  and  $b(x,y)$  are unwanted irradiance variations. In most cases  $\varphi(x,y)$ ,  $a(x,y)$  and  $b(x,y)$  vary slowly compared with the spatial-carrier frequency  $f_0$ . Then, the angle  $\varphi(x,y)$  is the phase shift caused by the object surface and the angle of projection, and its expressed as:

$$\varphi(x, y) = \varphi_0(x, y) + \varphi_z(x, y) \quad (2)$$

Where  $\varphi_0(x,y)$  is the phase caused by the angle of projection corresponding to the reference plane, and  $\varphi_z(x,y)$  is the phase caused by the object's height distribution.

Considering the figure 1, we have a fringe which is projected from the projector, the fringe reaches the object at point H and will cross the reference plane at the point C. The fringe then can be seen at the point F by the camera, and therefore the phase change caused by the object's shape is given by:

$$\varphi(x, y) = \varphi_0(x, y) + \varphi_z(x, y) \tag{3}$$

By observation, the triangles  $D_pHD_c$  and  $CHF$  are similar and

$$\frac{CD}{-h} = \frac{d_0}{l_0} \tag{4}$$

Leading us to the next equation:

$$\varphi_z(x, y) = \frac{h(x, y)2\pi f_0 d_0}{h(x, y) - l_0} \tag{5}$$

Where the value of  $h(x,y)$  is measured and considered as positive to the left side of the reference plane. The previous equation can be rearranged to express the height distribution as a function of the phase distribution:

$$h(x, y) = \frac{l_0 \phi_z(x, y)}{\phi_z(x, y) - 2\pi f_0 d_0} \tag{6}$$

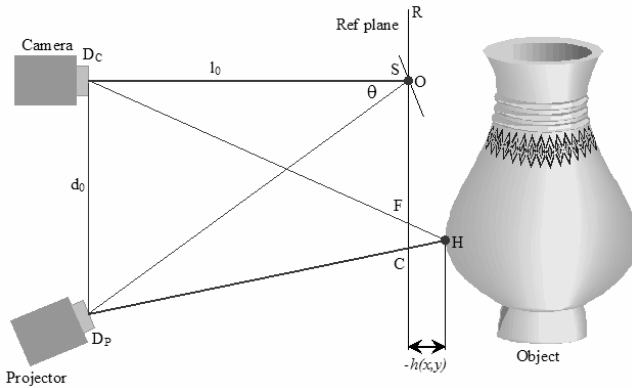


Fig. 1. Experimental setup

### 2.1 Fringe Analysis

The fringe projection equation 1 can be rewritten as:

$$g(x, y) = \sum_{n=-\infty}^{\infty} A_n r(x, y) \exp(in\varphi(x, y)) * \exp(i2\pi f_0 x) \tag{7}$$

Where  $r(x,y)$  is the reflectivity distribution on the diffuse object [3, 4]. Then, a FFT (Fast Fourier Transform) is applied to the signal for in the  $x$  direction only. Notice

that even  $y$  is considered as fix, the same procedure will be applied for the number of  $y$  lines in both images. Therefore, we obtain the next equation:

$$G(f, y) = \sum_{-\infty}^{\infty} Q_n(f - nf_0, y) \tag{8}$$

Now, we can observe that  $\varphi(x,y)$  and  $r(x,y)$  vary very slowly in comparison with the fringe spacing, then the  $Q$  peaks in the spectrum are separated each other. Also it is necessary to consider that if we choose a high spatial fringe pattern, the FFT will have a wider spacing among the frequencies. The next step is to remove all the signals with exception of the positive fundamental peak  $f_0$ . The obtained filtered image is then shifted by  $f_0$  and centered. Later, the IFFT (Inverse Fast Fourier Transform) is applied in the  $x$  direction only, same as the FFT. The obtained equations for the reference and the object are given by:

$$\hat{g}(x, y) = A_1 r(x, y) \exp\{i(2\pi f_0 x + \varphi(x, y))\} \tag{9}$$

$$\hat{g}_0(x, y) = A_1 r_0(x, y) \exp\{i(2\pi f_0 x + \varphi_0(x, y))\} \tag{10}$$

By multiplying the  $\hat{g}(x,y)$  with the conjugate of  $\hat{g}_0(x,y)$ , and separating the phase part of the result from the rest we obtain:

$$\begin{aligned} \varphi_z(x, y) &= \varphi(x, y) + \varphi_0(x, y) \\ &= \text{Im}\{\log(\hat{g}(x, y)\hat{g}_0^*(x, y))\} \end{aligned} \tag{11}$$

From the above equation, we can see that the phase map can be obtained by applying the same process for each horizontal line. The values of the phase map are wrapped at some specific values. Those phase values range between  $\pi$  and  $-\pi$ . Therefore, the next step is to apply some phase unwrapping algorithms. The whole methodology is described in figure 2.

The unwrapping consists of locating discontinuities of magnitude close to  $2\pi$ , and then depending on the phase change we can add or take  $2\pi$  according to the sign of the phase change. There are various methods for phase unwrapping, and the important thing to consider here is the abrupt phase changes in the neighbor pixels. There are a number of  $2\pi$  phase jumps between 2 successive wrapped phase values, and this number must be determined. This number depends on the spatial frequency of the fringe pattern projected at the beginning of the process.

This step is the modified part in the Fourier Transform Profilometry originally proposed by Takeda [3], and represents the major contribution of this work. Another thing to consider is to carry out a smoothing before the doing the phase unwrapping, this procedure will help to reduce the error produced by the unwanted jump variations in the wrapped phase map. Some similar methods are described in [5].



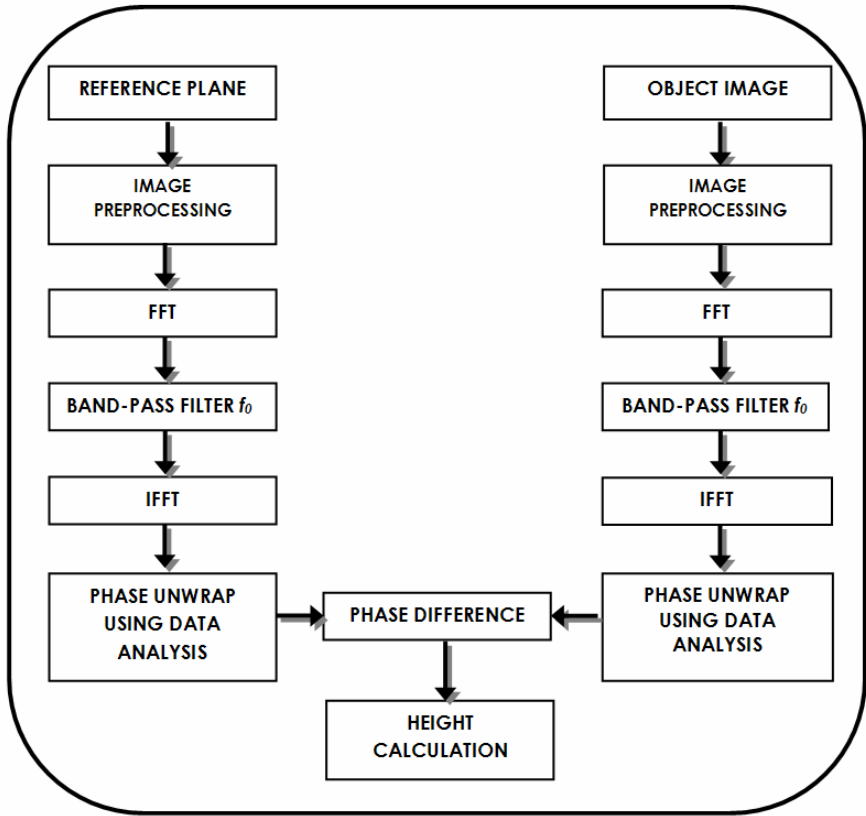


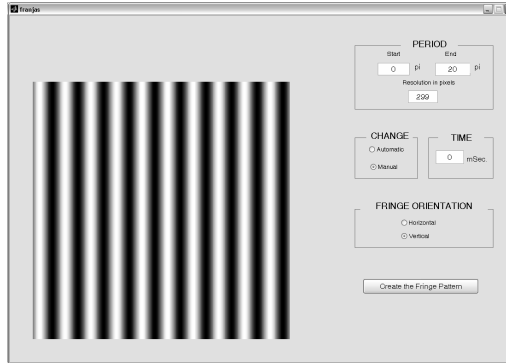
Fig. 2. Experimental setup

### 3 Experimental Results

To apply the proposed methodology, we implement an experimental setup like the one described in figure 1. A structured light fringe pattern is projected using a high resolution video projector and a digital CCD camera. As the reference plane we use a whiteboard covered with non-reflecting paper to avoid the unwanted reflections that affect the system. As an object, we tried 2 objects; a piece of metal with a triangular shape and a thermal coffee glass. To create a different fringe pattern, a GUI created in MATLAB was used like the one projected on figure 3. By using the GUI, we are able to modify the spatial frequency of the projected fringe pattern and the algorithm can be modified to use other well-known methods like phase shifting.

As an example of one object, we can see on figure 4 the reference pattern projected on a plane and the same pattern projected on the object.

Applying the modified Fourier Transform Profilometry we can obtain the Fourier spectra corresponding to the images on figure 5.



**Fig. 3.** Fringe Pattern GUI in MATLAB

On figure 6 we can observe the wrapped depth map before applying the unwrapped algorithm. Usually, in doing phase unwrapping, linear methods are used [5-7]. These methods fail due to the fact that in the wrapped direction of the phase, a high frequency can be present and a simple unwrapping algorithm can generate errors in the mentioned direction. Therefore, a more complete analysis should be carried out. In this work, a local discontinuity analysis and the use of a global analysis is proposed and implemented to solve the problem.

The algorithm for the local discontinuity analysis can be described as:

- The wrapped phase map is divided into regions with different weights ( $w_1, w_2, \dots, w_n$ ).
- A modulation unit is defined and helps to detect the fringe quality and divides the fringes into regions.
- The regions are grouped from the biggest to the smallest modulation value.
- The unwrapping process is started from the biggest to the smallest region.
- An evaluation of the phase changes can be carried out to avoid variations smaller than  $f_0$

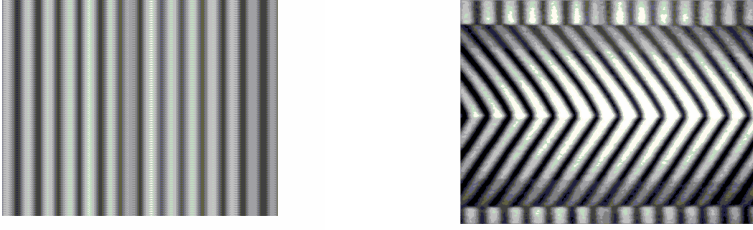
After the local analysis, an unwrapping algorithm is applied considering punctual variations in the phase difference image, which will lead us to the desired phase distribution profile, that is, the object's form.

Remember that all the phase unwrapping was carried out in the  $y$  direction.

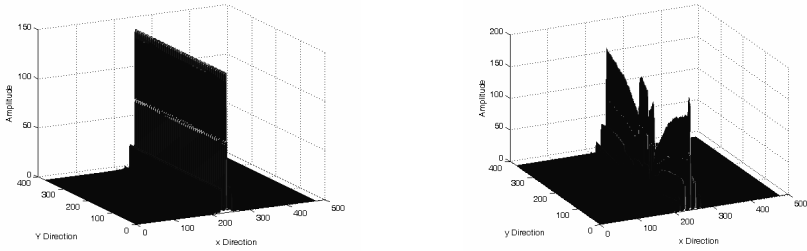
Here is also important to mention that all the programming was made in MATLAB.

Finally, on figure 7, the proposed method is applied to obtain the object's 3D reconstruction.

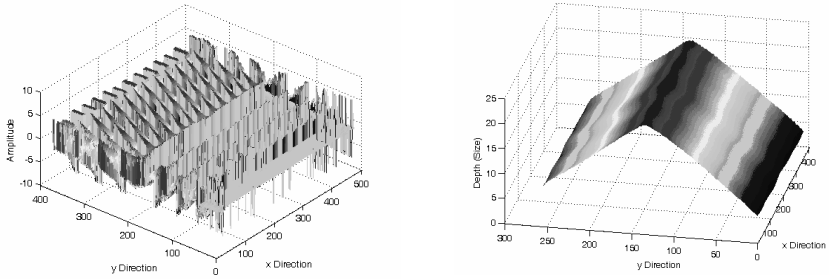
For the experimentation, we used a high resolution projector SONY XPL-CX5 (XGA), a CCD camera SONY TRV-30 1.3 Mega-pixels. As a reference frame, a wooden made plane was used, and it was painted with black opaque paint to avoid glare. The digitized objects were a metal piece and a cat ceramic face.



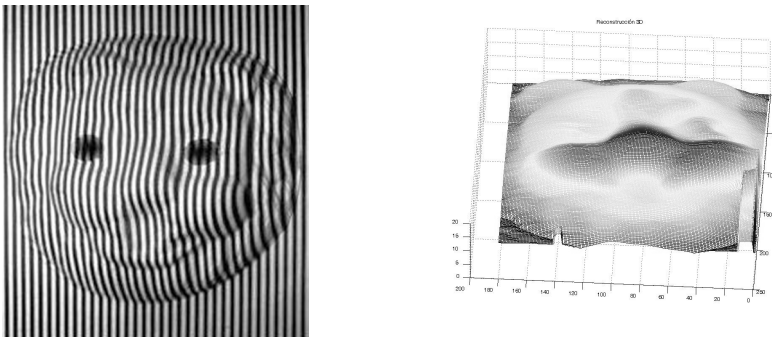
**Fig. 4.** Fringe Pattern projected on a plane and object to digitize respectively



**Fig. 5.** Fringe Pattern projected on a plane and object to digitize respectively



**Fig. 6.** Wrapped mesh and 3D reconstruction after unwrapping plus global and local analysis



**Fig. 7.** Object image and its 3D reconstruction after the proposed method is applied

## 4 Conclusions and Future Work

A modified Fourier Transform Profilometry method has been presented to improve the 3D reconstruction of objects, by modifying the unwrapping algorithms. The local and global analysis can be applied to the traditional FTP method to overcome the high frequency problem during the phase unwrapping step.

This kind of methodology could be used to digitize various objects in order to use them for future applications such as simulation, reverse engineering, virtual reality, 3D navigation depth map and so on.

A XY-R table is being constructed to obtain a full (360 degrees) 3D reconstruction.

Another challenge is to improve the speed of the algorithm in order to use the digitizer as close as possible to real time. One solution is to use an optical filter to obtain the FFT directly. Another solution is to implement the algorithm into a DSP or FPGA board. To improve the performance of the phase unwrapping, a wavelet or neural network approach is also considered for the future work.

Also there is the possibility of improve the results by applying an interpolation method like the splines or b-splines.

## References

1. Gokstorp, M.: Depth Computation in Robot Vision, Ph.D. Thesis, Department of Electrical Engineering, Linkoping University, S-581 83 Linkoping, Sweden (1995)
2. Pedraza-JC, Image Processing for Real World Representation Using Depth From Focus Criteria, Ph.D. Thesis, Department of Mechanical Engineering, University of Tsukuba (2002)
3. Takeda, M., Ina, H., Kobayashi, S.: Fourier-Transform method of fringe pattern analysis for computed-based topography and interferometry. *J.Opt. Soc.Am.* 72(1), 156–160 (1982)
4. Pynsent, F.B.P., Cubillo, J.: A theoretical Comparison of three fringe analysis methods for determining the three-dimensional shape of an object in the presence of noise. *Optics and Lasers in Engineering* 39, 35–50 (2003)
5. Rastogi, P.K.: Digital Speckle Pattern Interferometry and related Techniques. In: Edit, Wiley, Chichester (2001)
6. Itoh, K.: Analysis of the phase unwrapping algorithm. *Applied Optics* 21(14), 2470–2486
7. Lu, W.: Research and development of fringe projection-based methods in 3D shape reconstruction. *Journal of Zhejiang University SCIENCE A*, 1026–1036 (2006)

# 3D Space Representation by Evolutive Algorithms

Rodrigo Montúfar-Chavez<sup>1</sup> and Mónica Pérez-Meza<sup>2</sup>

<sup>1</sup> GIRATE Group, Engineering Division, ITESM Campus Santa Fe,  
Av. Carlos Lazo 100, Col. Santa Fe, Del. Álvaro Obregón  
01389 México D. F., México  
rmontufar@itesm.mx

<sup>2</sup> Universidad de la Sierra Sur,  
Guillermo Rojas Mijangos s/n, Ciudad Universitaria,  
70800 Miahuatlán de Porfirio Díaz, Oax., México  
mperez@unsis.edu.mx

**Abstract.** In this paper we present a system to obtain the representation of a 3D space using evolutive algorithms. Besides the evolutive algorithm, the proposed system is based on the mathematical principles of the vision stereo, particularly on stereoscopy. Vision stereo makes use of two images captured by a pair of cameras, in analogy to the mammalian vision system. Such images are employed to partially reconstruct the scene contained on them by some computational operations. In this work we employ only a camera, which is translated along a determined path, capturing the images every certain distance, providing the stereo images necessary for reconstruction. As we can not perform all computations required for the total scene reconstruction, we employ an evolutionary algorithm to partially reconstruct the scene and obtain its representation. The evolutive algorithm employed is the fly algorithm [1], which employ spatial points named “flies” to reconstruct the principal characteristics of the world following the rules of evolution dictated by the algorithm.

## 1 Introduction

Particularly in robotics, it is important the representation of the world where a robot is interacting because this representation provides safety to the robot during navigation. In this work we present a method to obtain 3D partial reconstruction of a scenario using stereoscopy, evolutive algorithms and monocular vision. The obtained reconstruction and some additional information provided by the robot sensors are enough to give security during navigation to mobile robots. Additionally, the representation is also useful for map construction tasks.

The artificial sense of depth and position in a scenario can be gotten by the fusion of stereoscopy with the fly algorithm, where spatial points are projected on a pair of displaced images and some computations are performed. The vision stereo system can capture the displaced images by one of the following procedures:

**Static capture:** In this procedure two or more cameras are separated one of each other certain distance (similar to mammalian vision system), and capture the images.

**Dynamic capture:** In this procedure one camera is used to get the set of images. It is displaced along a path, stopping every certain distance and capturing the images necessary for space reconstruction.

As mentioned above, in this work we use only one camera and dynamic capture. Once we have the stereo images, we perform the 3D scene reconstruction, applying geometry projective and using the fly algorithm to keep the most important or the best points in scene.

The system is considered to be employed in mobile robot navigation and map construction, applications where the robot needs to know every moment the structure of the world to move around in a secure way or construct the world.

## 2 Vision Stereo Principles

**Stereoscopy** [2, 3, 4] is a technique for inferring the three-dimensional position of spatial points in a scenario from two or more images. The pair of images, slightly displaced one of each other, has many characteristics in common, but they also have certain differences, and these differences are called *disparity*.

The reconstruction of the scene through vision stereo consists of two steps [5]: (1) *the correspondence problem* - for every point in one image find out the correspondent point on the other and compute the disparity of these points; and (2) *triangulation* - given the disparity map, the focal distance of the two cameras and the geometry of the stereo setting (relative position and orientation of the cameras) compute the 3D coordinates of all points in the images.

The design and the implementation of the stereo vision system must take into account two factors: (1) the correspondence problem and triangulation make the assumption to deal with an ideal model of the camera (pinhole model), that can be very different from actual imaging devices; and (2) the relative position and orientation of the two cameras must be known in order to retrieve the range information.

Therefore, the camera calibration [6, 7, 8] is a central issue for a stereo vision system. In fact, the calibration of a stereo camera is the task of relating the ideal pinhole model of the camera with the device employed (internal calibration) and retrieving its relative position and orientation (external calibration).

### 2.1 The Pin-Hole Model

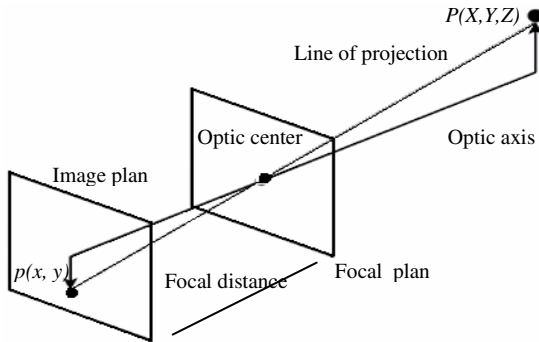
The pin-hole model camera is shown in Fig. 1. In this model, the three-dimensional point  $P(X, Y, Z)$  is projected on one image plan passing through the optic center located in the focal plan. The straight line that links the point  $P$  and the optic center is the projection line, and it intersects the image plan just in the pixel  $p(x, y)$ , which is the corresponding projection of  $P(X, Y, Z)$ .

The optic center is located on the center of the focal plan. This model is completed with the optic axis, which is a perpendicular line that begins in the center of the image plan, passing through the optic center and being perpendicular to  $P(X, Y, Z)$ .

According this model, and using the geometry rules, we obtain the equations that relate  $P(X, Y, Z)$  to  $p(x, y)$ .

$$x = f \frac{X}{Z} \quad \text{and} \quad y = f \frac{Y}{Z} \quad (1)$$

These equations are employed to project the population of flies we generate, on the images used for reconstruction.



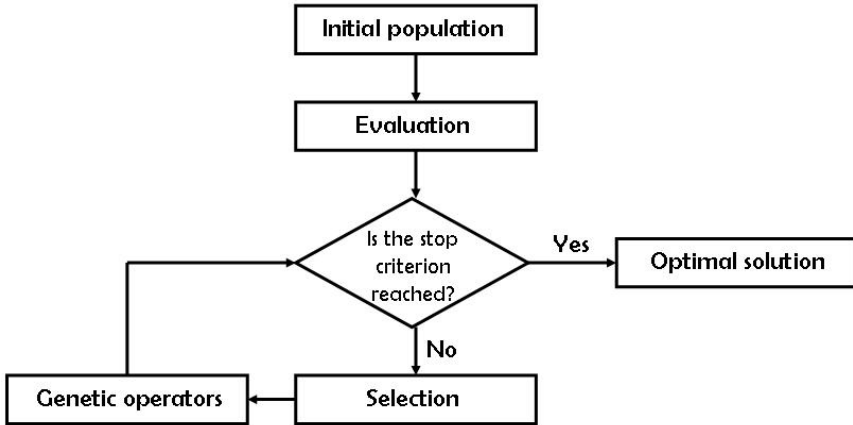
**Fig. 1.** The Pin-hole camera model. The spatial point  $P(X, Y, Z)$  is projected on the image plan, falling in the pixel  $p(x, y)$  of the image plan.

### 3 Evolutionary Algorithms

Evolutionary algorithms manipulate individuals, which are evaluated by a fitness function, in analogy to the biological evolution. The general schema of evolutionary algorithms is shown in Fig. 2.

The principal characteristics are:

- The population is a group of individuals.
- An individual is defined by genes  $X = (x_1, x_2, \dots, x_n)^T$ , which particularly represents its position  $(X, Y, Z)$  in the space.
- The evaluation is the computation of the fitness value in every individual.
- The selection eliminates part of the population, keeping the best individuals according to the evaluation.
- The evolution applies genetic operators (crossover, mutation, etc.), leading to generate new individuals in the population.
- Some kinds of evolutionary algorithms are:
- Genetic algorithms, which are a technique of programming that, imitate the biological evolution as a strategy to solve problems.
- Evolutionary strategies, which are rules that define the behavior of the individual under certain circumstances.
- Genetic programming, which are specific instructions in a programming language.
- Genetic algorithms evolve a population of individuals submitting it to random processes and actions, like in the biological evolution, and to a selection process according to certain criterion, where the most adapted individuals are selected to survive the process, and the less adapted are ruled out.



**Fig. 2.** General schema of the Genetic Algorithms. Population is evolving until best adapted individuals remain producing the optimal result.

### 3.1 The Fly Algorithm

The fly algorithm is considered an image processing technique based on the evolution of a population of flies (points in the space) projected over stereo images. The evolution is regulated by a fitness function determined in such way that the flies converge on the surface of an object located in the scene.

A fly is defined as a 3D point with world coordinates  $(X, Y, Z)$ . The flies are projected over a couple of despaiired images by stereoscopy, producing a pair of 2D coordinates:  $(x_R, y_R)$  and  $(x_L, y_L)$  for the right and left images respectively.

Initially, the population of flies is generated randomly in the intersection area of the view of both images and equally distributed on a certain number of regions to disperse all of them. The 3D space limits the range of the  $X$  and  $Y$  coordinates, meanwhile the  $Z$  coordinate ranges form 0.2 to 3 m. The fly algorithm considers the following functions:

**The fitness function.** The fitness function evaluates a fly, comparing projections on the left a right images. If a fly is located on an object, the projections will have similar pixel neighborhoods on both images and the fly will have a high fitness value. This idea is illustrated in Figs. 3 and 4. Figure 4 shows the neighborhoods of two flies on left and right images. In this example, *fly1*, which is located on an object, has a better fitness value than *fly2*. The fitness function  $F$  defined in [9, 10] is:

$$F = \frac{G}{\left( \sum_{colous} \sum_{(i,j) \in N} (L(x_L + i, y_L + j) - R(x_R + i, y_R + j))^2 \right)} \tag{2}$$

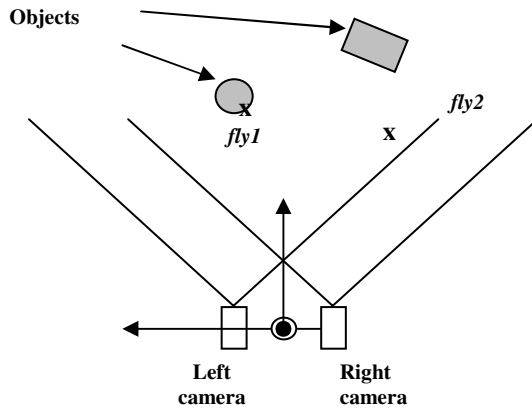
$$G = |\nabla(M_L)| |\nabla(M_R)| \tag{3}$$



where:

- $(x_L, y_L)$  and  $(x_R, y_R)$  are the pixel coordinates of the fly projected on the left and right images respectively.
- $L(x_L + i, y_L + j)$  and  $R(x_R + i, y_R + j)$  are the color values of the pixels in a neighborhood at left and right images respectively.
- $N$  is the dimension of the neighborhood population introduced to obtain a more discriminating comparison of the fly projections.

$|\nabla(M_L)|$  and  $|\nabla(M_R)|$  are the norms of the gradients of Sobel on the left and right projections of the fly. That is intended to penalize flies when they are located on uniform regions.



**Fig. 3.** View of two flies in the scene. The flies are projected on left and right image plans, evaluated and manipulated following the evolution rules of the fly algorithm.

In color images, the difference of squares in (2) is computed on each color channel. In gray images it is computed for one channel.

**Selection.** Selection is elitist and deterministic. It classifies flies according their fitness values and keeps the best individuals. A sharing operator [9, 10] reduces the fitness of clustered flies packed and forces them to explore other areas on the world.

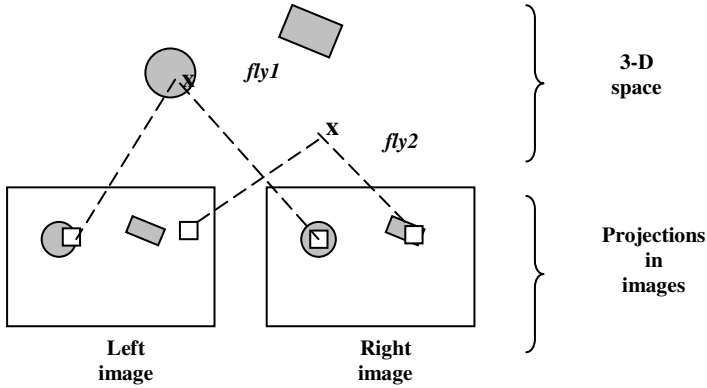
**Genetic operators.** We apply a pair of genetic operators to selected individuals.

*Gaussian mutation.* A new fly is generated adding Gaussian noise to each coordinate of the parent fly. We employ the  $\mathcal{N}(0, 1)$  distribution to generate the noise component for every coordinate, the random value generated is scaled according the vision field dimensions and then it is added to the respective coordinate.

*Barycentric cross-over.* A new individual is generated from two parents  $F_1$  and  $F_2$ , this individual is positioned between them, following:

$$\overline{F} = \lambda \overline{F}_1 + (1 - \lambda) \overline{F}_2 \tag{4}$$

where  $\lambda$  is a random value uniformly distributed between  $[0, 1]$ . In brief, the new individual is positioned on the imaginary line that connects  $F_1$  to  $F_2$ ,  $\lambda$  indicates the distance between the child and the parent  $F_1$ .



**Fig. 4.** Flies projections on the left and right images

## 4 Morphological Operators

We have incorporated some image processing operators to enhance the performance of the fly algorithm and the visualization of results. The included processes are edge detection, opening and closing operators.

### 4.1 Edge Detection

Edge detection is used to locate points where a sharp intensity variation is presented. The basic solution for many edge detection algorithms is the computation of local differential operators. We employ the Sobel operator [11] in the fitness function.

The Sobel operator measures the 2D spatial gradient on an image, emphasizing the regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in a grayscale image.

The Sobel operator consists of a pair of  $3 \times 3$  convolution kernels as shown in Eq. (5). One kernel is simply the other rotated by  $90^\circ$ . This is very similar to the Roberts Cross operator [12].

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{5}$$

## 4.2 Morphological Operator

Morphology is used in image processing to determine the shape or changes in an image. In this work, we employ the morphological operations dilation, erosion, opening and closing [11] to enhance the visualization of results.

**Dilation** ( $\oplus$ ). It is a morphological transformation that combines two pixel sets using the addition of vectors. The dilation of  $A$  by  $B$ , denoted  $A \oplus B$ , is defined as:

$$A \oplus B = \{x \mid (\hat{B})_x \cap A \neq \emptyset\} \quad (6)$$

$$\hat{B} = \{x \mid x = -b, b \in B\} \quad (7)$$

$$(B)_x = \{c \mid c = b + x, b \in B\} \quad (8)$$

**Erosion** ( $\otimes$ ). It is a morphological transformation that combines two pixel sets using the subtraction of vectors. It is the dual of dilation. Neither the erosion nor dilation is an inverse transformation.

$$A \otimes B = \{x \mid (B)_x \subseteq A\} \quad (9)$$

**Opening.** It is a two steps operation: the erosion followed by the dilation. This operator smoothes the contour of an image, breaking out narrow isthmuses and eliminating thin protuberances.

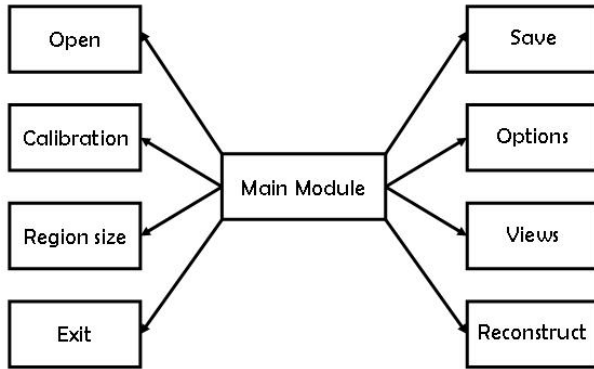
**Closing.** It is performed carrying out the dilation operation followed by the erosion operation. This operation tends to smooth sections of contours but, in opposition to the opening, generally fuses narrow separate and thin coming and deep, it eliminates small holes and fills holes of a contour.

## 5 3D Reconstruction System

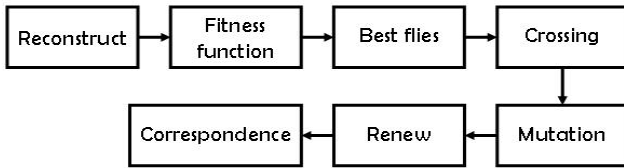
The developed system for reconstruction, shown in Fig. 5, has eight basic modules: Open, Save, Exit, Calibration, Region Size, Options, Views and Reconstruct. Additionally, the Reconstruct module, shown in Fig. 6, has five sub-modules: Fitness function, Correspondence, Crossing, Mutation and Renew.

The system modules are:

- a) *Open.* It opens a sequence of displaced images when camera is not available.
- b) *Save.* It allows saving images on screen.
- c) *Exit.* It exits the system.
- d) *Calibration.* The external and internal calibration parameters of the camera are introduced in this module.
- e) *Regions Size.* In this module we introduce the dimension of regions to divide the images. The idea of divide the images is avoid most of the flies converge in the same region or area. In this way, we disperse fairly the flies in the entire image.



**Fig. 5.** The system architecture for 3D representation. It consists of eight modules, each for a specific task.



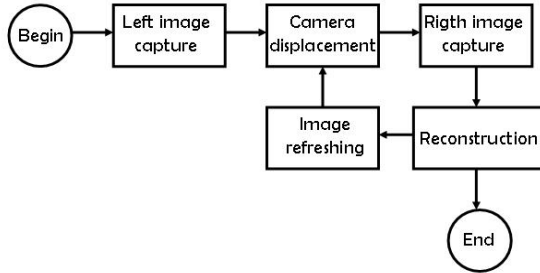
**Fig. 6.** The Reconstruct module. All reconstruction operations are carried out in this module. The flies are evaluated and the population refreshed and projected continuously.

- f) *Options.* The population size is introduced here.
- g) *Reconstruction.* This is the most important module. It performs the 3D reconstruction from the sequence of images. It has the following sub-modules.
  - i) *Fitness function.* The fitness function is applied to the population of flies. If the color differences between each fly and its neighborhood are low, almost zero, the value of the fitness function is high for this fly and vice versa.
  - ii) *Correspondence.* In this module we obtain the correspondence of the flies projected on the right and left images by stereoscopy.
  - iii) *Crossing.* This module is in charge of generating new flies from two parents using the crossing function.
  - iv) *Mutation.* In this module new flies are obtained by adding Gaussian noise to parent flies.
  - v) *Renew.* New flies are generated in this module.

The system is iterative, in such way that to carry out the reconstruction, the fitness function is applied every time. Once the fitness function has been computed for all individuals, we keep the best flies and delete the worse ones. The worse flies are eliminated when the operators of mutation, crossing and renew are applied. These operators are applied at different percentages of population previous to be projected on the right and left images.

As flies are displayed, we applied the morphological operators to the best individuals, obtaining the best representation of the 3D space reconstruction.

The general system process is presented in Fig. 7. As mentioned above, it is iterative, capturing continuously images, carrying out the reconstruction and refreshing or renewing the stereo images.



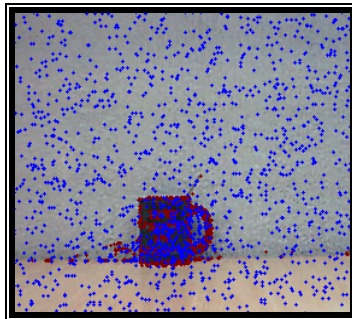
**Fig. 7.** The 3D reconstruction process. The system iteratively captures the stereo images, carry out the reconstruction and refresh the images.

## 6 Results

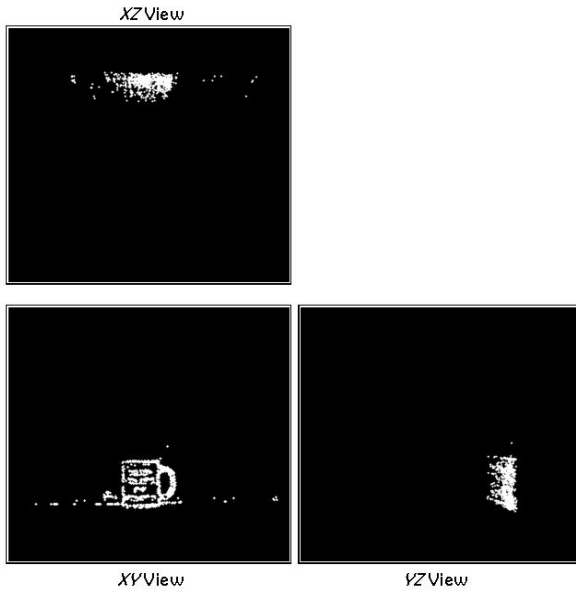
We present the results obtained with a population of 3000 flies, using a set of images displaced on horizontal axis two inches one from other. The left and right images are refreshed every two seconds approximately. The parameters employed in the fly algorithm are:

- Flies preserved every generation: 50%
- Flies generated by crossing: 20 %
- Flies generated by mutation: 20%
- Flies randomly generated: 10%

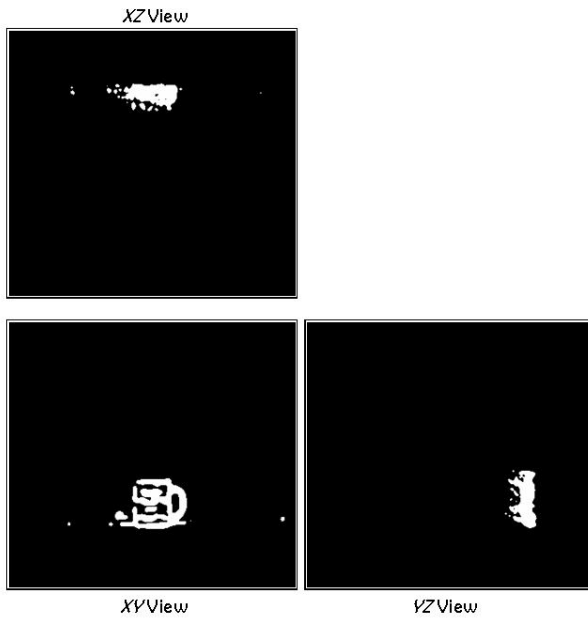
The images were divided in four regions to avoid clusters with a big quantity of flies. The population was fairly distributed, having 750 flies in every region.



**Fig. 8.** The flies positioning after 62 iterations and five stereo images. The best flies are in red on the object and the rest flies are in blue.

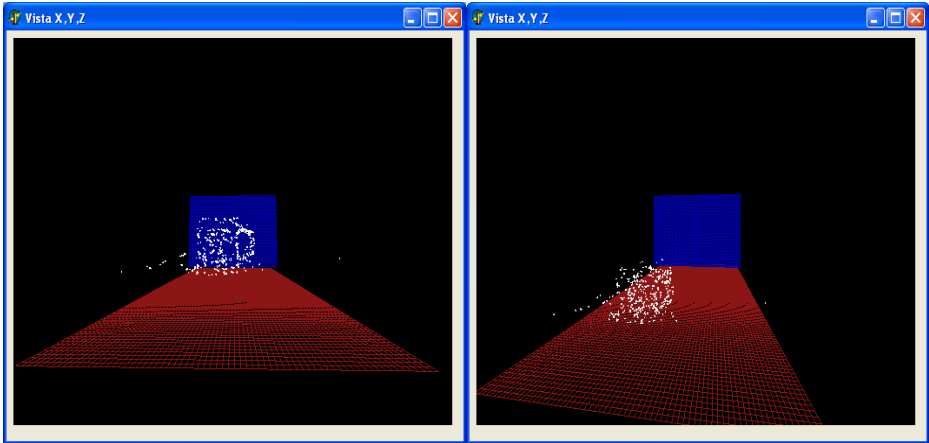


**Fig. 9.** The best flies visualization on the different views. The fusion of these views produces the 3D representation of the scene.



**Fig. 10.** The best flies visualization on the different views after applying the morphological operators. We have a best interpretation of the reconstruction.

Figure 8 shows the results after 62 iterations and five stereo images. The best flies are in red, located on the object and the rest of the population is in blue. Figure 9 presents the view of the three plans:  $XY$ ,  $XZ$  and  $YZ$ . We appreciate the flies giving dimension to the object. Figure 10 presents the same views after applying the morphological operators, producing a better interpretation of the object. Figure 11 shows the three dimensional representation from two perspectives.



**Fig. 11.** The three dimensional representation of the scene from two perspectives

## 7 Conclusions

In this work we have presented a system for partial scene reconstruction based on the fly algorithm and stereoscopy. The system produces a three dimensional representation very useful in robotics, especially for navigation and map reconstruction tasks. We have used monocular vision instead of stereo vision; this means only one camera is employed to capture the pair of stereo images. The camera displacement is carefully controlled.

The system allows setting easily all parameters involved in the process: the camera and the genetic algorithm parameters. Indoor several tests have been performed and the system works very well and the 3D reconstruction is obtained in a reasonable time for a mobile robot.

Images are divided in regions to disperse the flies on the entire image plan. In this way, we avoid an excessive concentration of flies in certain points.

Finally, we perform an opening and closing morphological operations to have a better appreciation of the reconstruction.

## References

1. Louchet, J.: Stereo analysis using individual evolution strateg. International Conference on Pattern Recognition. Barcelona (September 2000)
2. Marr, D., Poggio, T.: Cooperative computation of stereo disparity. *Science* 194, 283–287 (1976)

3. Gutierrez, S., Marroquin, J.L.: Disparity estimation and reconstruction in stereo vision. Technical communication No. I-03-07/7-04-2003. CC/CIMAT, Mexico (2003)
4. Quam, L., Hannah, M.J.: Stanford automated photogrammetry research. Technical Report AIM-254, Stanford AI Lab (1974)
5. Marr, D., Poggio, T.: A Computational Theory of Human Stereo Vision. Proceedings of the Royal Society of London. Series B, Biological Series 204(1156), 301–328 (1979)
6. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(11), 1330–1334 (2000)
7. Abdel-Aziz, Y.I., Karara, H.M.: Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In: Proc. of the Symp. on Close-Range Photogrammetry, Falls Church, VA, pp. 1–18 (1971)
8. Tsai, R.Y.: A versatile camera calibration technique for 3D machine vision. *IEEE Journal of Robotics and Automation* RA-3(4), 323–344 (1987)
9. Boumaza, A.M., Louchet, J.: Dynamic Flies: Using Real-Time Parisian Evolution in Robotics. In: *EvoWorkshops*, pp. 288–297 (2001)
10. Louchet, J., Guyon, M., Lesot, M.J., Boumaza, A.: Dynamic Flies: a new pattern recognition tool applied to stereo sequence processing. *Pattern Recognition Letters* 23, 335–345 (2002)
11. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Addison-Wesley, Reading (1992)
12. Roberts, L.: *Machine Perception of 3-D Solids*. In: *Optical and Electro-optical Information Processing*, MIT Press, Cambridge (1965)



# Knowledge Acquisition and Automatic Generation of Rules for the Inference Machine CLIPS

Veronica E. Arriola<sup>1</sup> and Jesus Savage<sup>2</sup>

<sup>1</sup> Sciences Faculty, University of Mexico, UNAM  
varriola@ada.fciencias.unam.mx

<sup>2</sup> Laboratory of Biorobotics, University of Mexico, UNAM  
savage@servidor.unam.mx

**Abstract.** A hierarchical representation of objects is dynamically generated from the input of a virtual vision system. It is used to analyze a sequence of actions and extract behavior rules that can be utilized by the inference machine CLIPS. The vision system is assumed to provide simplified positional and shape information about visible 3D silhouettes in a frame per frame basis. A virtual agent, attempts to keep track of every image, without any previous knowledge about the object it represents. The hierarchy is restructured as necessary, to include new perceived images, in such a way that it also reflects factual relationships amongst them. Modifications between consecutive frames are internally interpreted and represented as functions which take the original world description and transform it into the next frame. A partial order is defined while looking for the satisfaction of domain/codomain requirements in functions composition, thus leading to the CLIPS rules.

## 1 Introduction

Currently there is a growing need to tackle the integration of perception and action, knowledge generation, dealing with context, novelty and allowing introspection. This need has been posed by numerous authors and projects, starting with Terry Winograd [1] back on the 70's and continuing with new projects like CoSy at Birmingham, where members like Aaron Sloman [2] and Hawes et al. [3] try "to construct physically instantiated or embodied systems that can perceive, understand ... and interact with their environment, and evolve in order to achieve human-like performance in activities requiring context ... specific knowledge". In the same attempt to look for this modules integration Savage, Billingham and Holden [4] have designed a whole agent architecture that specifies the interactions among modules, starting with the perception system, going through logical layers and leading to actions.

In this work, a dynamically generated knowledge representation and a set of algorithms were developed in order to allow a virtual agent to transform simple descriptions of visualized images into high level behavior rules for an inference

machine (CLIPS). Since the agent does not receive previous information about the images it observes, all its knowledge is acquired from its own experience, thus allowing it to give significance to things according to the context in which it knew of them.

The design is still simple, but it tackles the management of concepts difficult to acquire and it succeeds in acquiring them.

## 2 Visual Information

The input for the agent is provided through a vision module. This module abstracts the notion of a vision system with segmentation capabilities, so that it can describe perceived images in terms of the objects (or segments of objects) in them and their relative positions.

### 2.1 Requirements

It is assumed that a vision system provides the agent with descriptions of every observed image, in a frame per frame basis. Every description shall include the following information:

1. Image segmentation in simple 3D silhouettes. That is, it separates objects according to its 3-dimensional individuality.
2. Shape information. Currently this is limited to a preliminary grouping of similar images.
3. Positional information. At the moment, only adjacency is taken into account (two or more silhouettes are in contact or they are not)
4. Frame by frame images description.

### 2.2 Representation

A special notation, inspired in LISP syntax [5], was introduced to describe these images.

Every frame is represented by nested lists of atoms. Atoms correspond to minimal segments identified by the vision system. Elements inside a list correspond to segments (or groups of segments) in physical contact.

*Example 1* (Notation for descriptions of the images)

1. The image of a left hand:  
(mi)
2. Two hands and two feet:  
(mi md pi pd)
3. Two hands grasping a sock, both feet nearby:  
((mi cal md) pi pd)

The gradual transformation of several adjacent atomic segments into one is received (from the vision system) as a fusion of them and is represented by a right pointing arrow, from the original group at the previous frame to the atom representing the new shape. Its complementary operation (one atomic segment is split into many) would correspond to a fission and its notation is analogous.

*Example 2* (Notation for fusions and fissions of segments)

1. A sock is put on the right foot:

Frame 1: (mi (cal pd) md)

Frame 2: (mi ((cal pd) -> cpd) md)

2. The sock is taken off:

Frame 1: (mi cpd md)

Frame 2: (mi (cpd -> (cal pd)) md)

The required grouping by shape similarity is expressed as classes hierarchies. After all, the agent sees every different image as a definition for a new image class. Thus, similar images shall inherit from a common base class (multiple inheritance is allowed). This relationship is denoted by a double lined arrow pointing from the parent classes to the child.

*Example 3* (Notation for similar images)

1. The image of a right hand shall inherit from a hand image class:

(m => md)

2. The image of a left hand shall inherit from a hand image class:

(m => mi)

In this way, at every frame, lists describe which silhouettes are present, if they are adjacent and if some from a previous frame got fused or disengaged.

### 3 Dynamic Internal Generation of Hierarchies of Images' Classes

Every description of an atomic image has associated a unique class that represents it internally. For the case of a group there is a class that holds attribute references to the classes of its constituent elements (as many as necessary), which can be atomic or other groups' classes.

Every time the agent reads a frame description it registers any element it had not seen before. In the case of atoms the procedure is simple: if it is not found, it is added along with its parents (if that information is made available). However, for groups, more processing is required.

Groups with enough elements in common are considered alike, and this likeness is also reflected in the classes hierarchy. The criterion used is that two groups are considered alike if they share at least half of their elements. In order to reflect this at the classes hierarchy, every time a group description is read, the agent tries to determine if it has already registered it. If it is not the case, while looking for it, it keeps track of the most similar group. Once the search is over, if it found a most similar one which covers the similarity criterion, it generates a base class (or takes it as its base class, depending on the shared elements) which holds the common attributes. References from the new children are displaced to the parent as required so that they only hold non common attributes. Fig. 1 shows an example where a base class holds the references to the common elements of two group classes. The first group C1 can represent both hands holding the left foot and a sock, while C3 are the two hands holding only the sock.

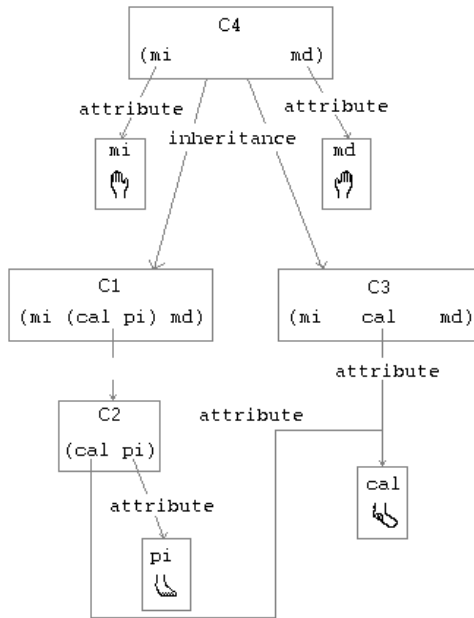


Fig. 1. The parent class C4 holds references to the common attributes of C1 and C3

By restructuring the hierarchy in this way and at the signaled times, it does not only reflect conformation similarities, but also, this similarities are registered as they are observed through time. During our experiment, this allowed to relate the image of hands holding a sock, or a shoe with those putting them on the foot. The links for coding the hierarchy are also used to speed searches while analyzing new images, since movements from time to time involve transforming images into similar ones. However, this algorithm still requires improvements, since it does not tackle complex cases.

## 4 Internal Instances

For this system, the image is only what the vision system perceives, it is only an accident of the object it conceives. Therefore, it requires a principle to determine how it can associate an image to the object it tries to follow. This principle is:

**Definition 1.** *When going from a frame to the next one (from a time  $t_1$  to a time  $t_2$ ) it must be considered that objects that look alike, and are found at a maximally close position to the position of its correspondent analogous at the previous frame, are the same, unless it has been explicitly declared otherwise.*

### 4.1 Tracking of Instances Through Time

According to this, when the agent begins its analysis of a sequence, it generates an internal representation where one object is associated per atomic image and a group object (with references to its elements) per group are created. From then on, it will try to repeat associations with the already created objects for the images on the next frame. Changes from frame to frame will be interpreted with the aid of the next basic transformations:

1. Appear: An image(atomic segment or group) is seen at a position where it was not before.
2. Disappear: An image is not seen where it was.
3. Merge: The fusion of adjacent atomic segments.
4. Split: The fission of an atomic element into many.

Once the objects at the frame have been recognized, thanks to the class system, only groups or atoms which where modified will require further analysis. Initially, these differences will be recorded as sets of the four basic transformations. The next step consists in relating images that disappeared at the previous frame with identical images that appeared at the next one, interpreting this as movements of the object. Only in cases where these associations are not possible they will be considered as apparitions of disappearances of objects.

For the next example, instances will be denoted by its image class name followed by a number inside brackets. This number allows to distinguish among distinct instances that look the same.

*Example 4* (Following instances through time)

Internal representation of the previous frame:

```
(foo[7] C#0[0] pi[0] C#1[3] zi[2] something[9] cal[0])
```

with the groups:

```
C#0[0] = (md[0] C#3[0] mi[0])
```

```
C#3[0] = (cal[1] pd[0])
```

```
C#1[3] = (cal[2] zd[2])
```

Internal representation of the current frame:

```
(foo[7] C#5[0] pi[0] cal[2] zd[2] zi[2] something[9] cal[0])
```

with the groups:

```
C#5[0] = (md[0] cpd[0] mi[0])
```

The transformations are:

```
MOVE: cal[2] TO top
```

```
MOVE: zd[2] TO top
```

```
MERGE: C#3[0]
```

This frames could be interpreted as the description of a very disordered bedroom, with some unidentified objects, identical socks (cal), shoes (zd, zi), hands (md, mi) and feet (pi, pd). What happens between frames is that a sock falls from a right shoe, while a person puts on another one.

## 5 Learning

Once the perceived images have been analyzed and the internal representation of the world has been modified in accordance, the agent proceeds to analyze the properties of the sequence of modifications. Behavioral rules will be extracted from this analysis.

### 5.1 From Frame to Frame

The first step for learning is to generalize the way frame to frame actions are stored. Instead of making reference to concrete instances, new functions will be declared that can generally transform any internal representation with the required objects present into the next described frame. This functions will take the positions of the original objects described in terms of its type. That is, the functions factory takes a list of the probably nested groups inside of which the displaced object was (this object can be an atom or a group) until the final member in the list is the class of the object itself and the position to where it is moved (described also in terms of types). The new function will receive the frame description as its parameter and it will look for an object at the previously indicated position, then it will transform the internal representation of the world so that the chosen object is moved to the previously identified new position.

The entire frame to frame transformation will be accomplished by the application of the sequence of individual registered transformations. Now every time the same type of differences are detected among frames, the transformations analysis is not repeated. The already created function is invoked over the internal representation. This functions are stored and indexed by the listing of the initial participating elements and their final states (only the first level classes names are required) thus, fastening the access to them for reuse in latter analysis.

*Example 5* (Frame to frame transformation function). Following with the previous example, the corresponding function description is:

```
PARTICIPANTS: C#1 C#0
SEQUENCE:
f1: MOVE (C#1 cal) TO top
f2: MOVE (C#1 zd) TO top
f3: MERGE (C#0 C#3) INTO cpd
```

```
FINAL ELEMENTS: C#5 cal zd
```

Note: in the case of merge the resulting atom shall be in the same position of the original group, therefrom it suffices to specify the class of the new atom.

## 5.2 Sequence Analysis

Now that the entire frame transformation can be characterized by a tuple with the initial and final state descriptions, the analysis of the sequence can be done at a higher level of abstraction.

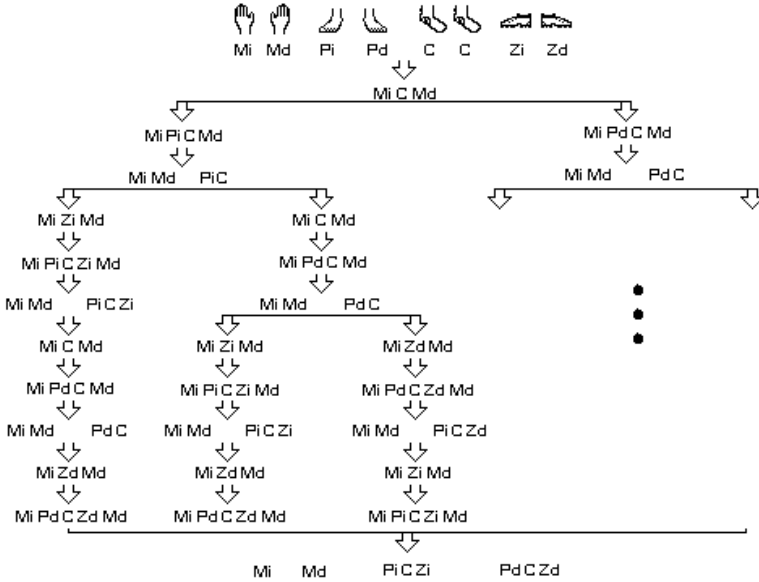
The final part makes use of a very simple principle for functions composition: in order for a function to be composed with other, the codomain of the first one must be contained by the range of the second one; otherwise the second one could not be applied. In this case, the essence of this principle is taken: in order to apply one of the transformations, the state in which the previous one left the world should still cover the requirements of the next one. Concretely, the required participating objects should be available.

What the system currently does is, once finished the observation and frame by frame analysis of the sequence, it takes all the frame to frame functions and generates a tree of all the possible orders in which they could be applied and lead to the final desired state. Fig. 2 shows a partial tree generated from the required actions to put on socks and shoes.

From this tree, the agent groups together all of those transformations whose order could never be altered, defining a new cluster of ordered actions, also characterized by its initial and final states descriptions, as shown in Fig. 3.

By traversing the tree from top to bottom, through any of its branches, it can reproduce the required actions to reach the final state. It suffices now to translate this information to CLIPS syntax [6] and the program is ready to be used. The names of the images classes are used for CLIPS rules labels, and the clusters of functions initial and final states establish the left and right elements of the rules.

The next example shows the generated code for our experiment. The structure is still complex, it also includes a group class, corresponding to both hands holding the sock, while any human made code thought of it. The reason was that, after grasping the first sock, it is still possible to decide on which foot it will be put. There are still missing symmetries, the rules were never simplified taking into account the similarities between dressing the left and the right sides; but this is natural, since the system has still very limited information management and analysis capabilities.



**Fig. 2.** Tree that shows three of the six different sequences that can be used to put on socks and shoes

*Example 6 (Generated CLIPS rules)*

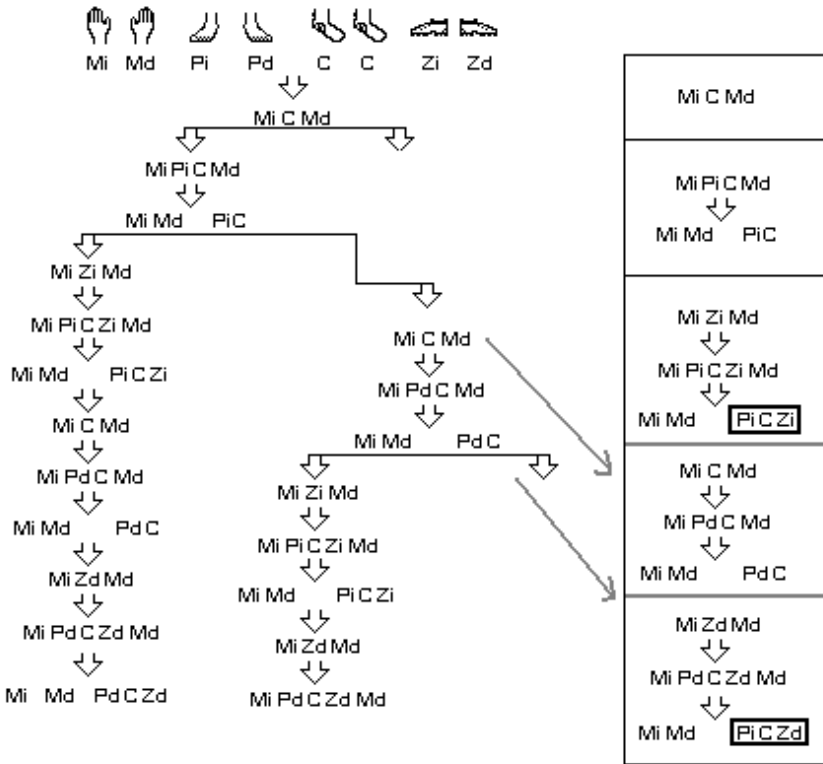
```
;;
;; CODE GENERATED BY THE AGENT
;;
;;Participating objects
(deftemplate ZCP (slot hijo))
(deftemplate ZAPATO (slot hijo))
(deftemplate CP (slot hijo))
(deftemplate PIE (slot hijo))
(deftemplate C#248 (slot hijo))
(deftemplate MANO (slot hijo))
(deftemplate CAL (slot instancia))

;;Initial state
(deffacts estado-inicial
  (CAL (instancia 1)) (CAL (instancia 2))
  (PIE (hijo PD)) (PIE (hijo PI))
  (ZAPATO (hijo ZD)) (ZAPATO (hijo ZI))
  (MANO (hijo MD)) (MANO (hijo MI))
  (META Ponerse los zapatos))
```



```
;;Actions
(defrule accion1 (META Ponerse los zapatos)
  ?f1<- (CAL (instancia ?instancia))
  ?f2<- (MANO (hijo MD))
  ?f3<- (MANO (hijo MI))
  =>
  (retract ?f3 ?f2 ?f1)
  (assert (C#248 (hijo C#246))))
(defrule accion2 (META Ponerse los zapatos)
  ?f1<- (C#248 (hijo C#246))
  ?f2<- (PIE (hijo PD))
  =>
  (retract ?f2 ?f1)
  (assert (CP (hijo CPD)))
  (assert (MANO (hijo MD)))
  (assert (MANO (hijo MI))))
(defrule accion3 (META Ponerse los zapatos)
  ?f1<- (CP (hijo CPD))
  ?f2<- (ZAPATO (hijo ZD))
  =>
  (retract ?f2 ?f1)
  (assert (ZCP (hijo ZCPD))))
(defrule accion4 (META Ponerse los zapatos)
  ?f1<- (C#248 (hijo C#246))
  ?f2<- (PIE (hijo PI))
  =>
  (retract ?f2 ?f1)
  (assert (CP (hijo CPI)))
  (assert (MANO (hijo MD)))
  (assert (MANO (hijo MI))))
(defrule accion5 (META Ponerse los zapatos)
  ?f1<- (CP (hijo CPI))
  ?f2<- (ZAPATO (hijo ZI))
  =>
  (retract ?f2 ?f1)
  (assert (ZCP (hijo ZCPI))))

;;End
(defrule termina
  ?f0 <- (META Ponerse los zapatos)
  (ZCP (hijo ZCPI)) (ZCP (hijo ZCPD))
  =>
  (retract ?f0))
```



**Fig. 3.** While traversing the tree, the agent detects branching points and uses them to define order segmentations of the sequence

## 6 Conclusions

By chaining sets of functional clustering analysis it was possible to move from very granular images descriptions sequences to the high level required rules for an inference machine. As mentioned before, the applied criteria and algorithms only cover the simpler cases. Since the algorithms are deterministic, and the describable images too simple, the outcome of alternative experiments is predictable. However the method can be extended to cover complexer cases. The general idea could still be same, even though every sub-algorithm will have to be greatly extended to allow the incorporation of non symbolic information and richer objects descriptions.

A more detailed comparison between the generated code and one made by human, reveals more aspects that the agent is not considering yet. These points are nothing more that the tip of the iceberg. As stated from the very beginning, this experiment was done as a first step towards learning abstract concepts from pure perceptual experience, but there is still a long way to go.

The short space available does not allow a deeper analysis here but more details of this work can be read at [7].

## References

1. Winograd, T.: Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. MIT AI Technical Report 235 (1971)
2. Sloman, A.: AI in a New Millenium: Obstacles and Opportunities. Birmingham CoSy Technical Reports (2005)
3. Hawes, N., Sloman, A., Wyatt, J., Jacobsson, H., et al.: Towards an Integrated Robot with Multiple Cognitive Functions. In: Forthcoming in the AAAI 2007 special track on Integrated Intelligence (2007)
4. Savage, J., Billingham, M., Holden, A.: The Virbot: A Virtual Reality Mobile Robot Driven with Multimodal Comands. *Expert Systems with Applications* 15, 413–419 (1998)
5. McCarthy, J.: Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I. *Communications of the ACM (CACM)*, 184–195 (April 1960)
6. Giarratano, J.C.: CLIPS: User Guide. Version 6.20 (2002), <http://www.ghg.net/clips/download/documentation/usrguide.pdf>
7. Arriola, V.: Generacion Automatica de Reglas para la Maquina de Inferencias CLIPS. Thesis for the Master Degree in Computer Sciences, Mentor: Savage, J., UNAM (2006)

# On-Line Rectification of Sport Sequences with Moving Cameras

Jean-Bernard Hayet<sup>1,\*</sup> and Justus Piater<sup>2</sup>

<sup>1</sup> CIMAT, A.C., Jalisco S/N  
36240 Guanajuato, GTO., Mexico

<sup>2</sup> Institut Montefiore, University of Liege  
4000 Liege, Belgium

**Abstract.** This article proposes a global approach to the rectification of sport sequences, to estimate the mapping from the video images to the terrain in the ground plane without using position sensors on the TV camera. Our strategy relies on three complementary techniques: (1) initial homography estimation using line-feature matching, (2) homography estimation with line-feature tracking, and (3) incremental homography estimation through point-feature tracking. Together, they allow continuous homography estimation over time, even during periods where the video does not contain sufficient line features to determine the homography from scratch. We illustrate the complementarity of the 3 techniques on a set of challenging examples.

## 1 Introduction

In the current era of mass entertainment, sport broadcasting has become an indispensable ingredient. Given the interests at stake and the huge demand for game analysis, much research has been done over the last decade for enhancing the broadcast video data with meta-data of particular interest to sports fans or coaches, such as player trajectories, off-side lines, etc. Ideally, these data should be produced instantly to help to understand the game as it evolves. Of course, computer vision is at the heart of this research effort, since it can provide the needed automatic procedures, which are usually done in a labor-intensive way. In terms of metric concepts, one needs to transform the relevant data defined in the image coordinate system into a real-world coordinate system, in a frame attached to the terrain field, a process generally referred to as “rectification”.

For planar scenes, the image-to-model mapping is a homography, or linear mapping in  $\mathbb{P}^2$  [1]. It depends on the current internal configuration of the camera and its position with respect to the field. In particular, under camera motion, this transformation evolves. In the case of sport scenes observed by static cameras, one may perform classical calibration beforehand, and then use the computed homography [2]. Now, when the camera moves (as in most outdoor

---

\* This work was sponsored by the Region Wallonne under DGTRE/WIST contract 031/5439.

sports), one needs to estimate this transformation continuously, which makes the pre-calibrated framework infeasible. Another key requirement of a scene analysis system is the management of uncertainties, as it is important to evaluate the precision of a given position or velocity, e.g. in multi-camera tracking applications. This article proposes a global approach to estimating the image-to-scene homography and its uncertainty in most typical team-sports scenarios, without camera motion sensor.

Most previous work do not fully exploit the temporal continuity of the image sequence and rely on pattern recognition techniques based on line-feature matching [3,4]. In some cases, we may try to calibrate the camera entirely by using the geometric properties of the scene [5]. However, this is often unnecessary, unless the application intrinsically requires 3D information, e.g. for the ball position [6]. In the case of ice hockey, an interesting approach was proposed that involves tracking points within the video over time to estimate inter-image homographies, and using line and circle features for fitting the field model [7]. However, such an approach is difficultly adaptable to soccer, as line features are not always present. In the context of sports, inter-image homographies have been used in mosaicking applications [8]; one of our ideas is to accumulate them across frames to provide estimates of the image-to-model transformation.

The transformation between a plane and its projection is well known as a homography. Here, the image-to-model homography maps points in the model to points in the image through the  $3 \times 3$  matrix  $H$ , that is, up to a scale factor,

$$HP \sim p, \quad (1)$$

where  $P = (X, Y, 1)^T$  is a model point and  $p$ , image of the point  $P$  through the TV camera, is denoted as  $(u, v, w)^T$ . As an input to our system, we consider a model (e.g., the soccer rules) composed of  $N$  line segments  $\mathcal{M} = \{S_k = (P_k^b, P_k^e), k = 1, \dots, N_M\}$ , where  $S_k$  is a line segment with vertices  $P_k^b$  and  $P_k^e$  and support line  $L_k \in \mathbb{R}^3$ , i.e.  $(L_k)^T P = 0$  if point  $P$  belongs to  $L_k$ .

An overview of our rectification approach is depicted in Fig. 1. Three modules are used for initializing and maintaining the homographies. The line-detection (LD) module allows for the initialization of the homographies from scratch, as shown in light gray in Fig. 1. This module is based on the approach presented in [4], which relies on a geometrically-motivated hypothesis generation-verification scheme to match the set of detected lines (i.e., white markings on football fields) to a known model. Although the vanishing points detection is not very precise, we use the approximate results given by this algorithm as an input of the two other modules. The line-tracking (LT) module, shown in medium gray, tracks the lines whose positions can be currently estimated using classical registration techniques. A module for ego-motion estimation, and thus called the visual-odometry (VO) module, shown in dark gray, incrementally computes the motion across images by tracking feature points from image to image. The LT and VO modules are successively described in the next sections.

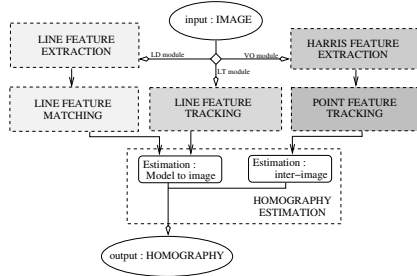


Fig. 1. Overview of our approach

## 2 Line Tracking (LT): Updating of Homographies

When a sufficient number of line features are visible, efficient and precise line tracking from image to image is critical for ensuring fast re-estimation of the homography  $H^t$ , i.e. the image-to-model homography at time  $t$ , supposing we have an estimation of  $H^{t-1}$ . This section briefly describes how line tracking is performed and explains how its results are used to estimate  $H^t$ .

Suppose that we have estimated  $H^{t-1}$  at time  $t-1$ . Then, all the line segments that form the field model  $\mathcal{M} = \{S_k = (P_k^b, P_k^e), k = 1..N_M\}$  can be reprojected onto the image in a set  $\{s_k = (p_k^b, p_k^e), k = 1..n_M\}$ , where  $n_M \leq N_M$ . Each of these segments  $s_k$  is warped at regular samples along the direction that is orthogonal to the line. We use for that a traditional correlation technique on color profiles [9], i.e., at every point  $\pi_{k,l}$ , we look for a point  $\pi_{k,l}^*$  such that the correlation with a reference color profile is maximum in the search direction.

These maxima are used as candidate points for the estimation of the new line parameters for the corrected version  $s_k^*$  of  $s_k$  through a robust RANSAC procedure on the set of  $\{\pi_{k,l}^*\}$ . It allows to get rid of outliers due to spurious local warpings, e.g. coming from players. The result of this process is a set of line segments  $s_k^*$  together with the corresponding inliers  $\pi_{k,l}^*$ .

If a sufficient number of matches are available between the current image and the field model, then we can use them to estimate the homography  $H$ .

In the classical approach, one would use line correspondences between  $s_k^*$  (in the image) and their counterpart in the model, the line segments  $S_k^*$ . Two constraints on the homography  $H$  would be established for each correspondence; however, the line parameters do not constitute very reliable measures, as a first estimation of them is needed. By contrast, we use the points  $\pi_{k,l}^*$  that have been successively warped to express constraints on the homography  $H$ :

$$L_k^T(H\pi_{k,l}^*) = 0. \tag{2}$$

This equation is asymmetric and differs from traditional approaches that use direct correspondences (points-to-points or lines-to-lines). Here, each of the  $n_s$  warped point gives rise to one linear equation (instead of two) in the homography,

so that we obtain a linear equation in the  $9 \times 1$  vector  $h$  containing the elements of the matrix  $H$ , i.e.  $h = (H_{11}, H_{12}, H_{13}, \dots, H_{33})^T$ .

The complete system has the form  $Sh = 0$ , where  $S$  is a  $n_s \times 9$  matrix. Each row  $s_i$  of the matrix  $S$  is given by

$$s_i = (\sin \alpha_{\iota(i)} u_i \sin \alpha_{\iota(i)} v_i \sin \alpha_{\iota(i)} \cos \alpha_{\iota(i)} u_i \cos \alpha_{\iota(i)} v_i \cos \alpha_{\iota(i)} c_{\iota(i)} u_i c_{\iota(i)} v_i c_{\iota(i)})$$

where  $(u_i, v_i)$  are the coordinates of one of the warped point, and  $L_k = (\sin \alpha_k \cos \alpha_k c_k)^T$  are the  $k$ -th line parameters. We denote by  $\iota(i)$  the mapping that associates an index among the rows of matrix  $S$  with an index among the set of lines, i.e.,  $\iota(i) = \iota(j)$  when the equations coming from rows  $i$  and  $j$  arise from points belonging to the same line. The solution of this system is clearly the right singular vector of matrix  $S$  associated to the smallest singular value. Note that, since the points have been filtered through local RANSAC on each warped line, there is no need for a robust strategy here; just one linear system is solved.

Moreover, to improve the precision of the linear rectification method presented above, we perform a few iterations of non-linear optimization in a Levenberg-Marquardt scheme. The geometric criterion  $C$  we minimize is the sum of the squared distances between warped points and the projection of their corresponding line in the model, i.e.

$$C = \sum_{k,l} (n(H^{tT} L_k) \cdot \pi_{k,l}^*)^2$$

where  $n(l) = \frac{1}{\sqrt{l_1^2 + l_2^2}}(l_1, l_2, l_3)^T$  for  $l \in \mathbb{R}^3$ ,  $(l_1, l_2) \neq (0, 0)$ . Figure 2 shows examples where the optimization of  $C$  in  $\mathbb{R}^9$  improves the estimation of  $H$ .



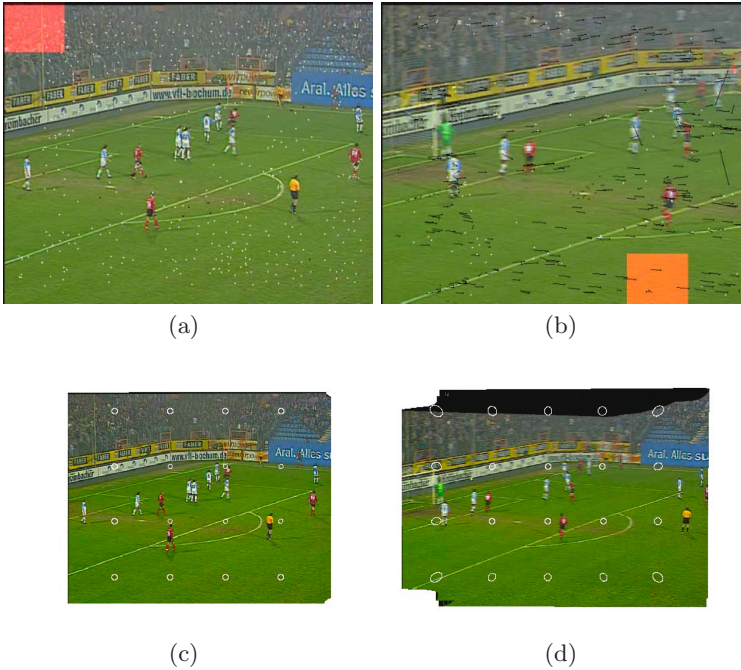
**Fig. 2.** Effects of non-linear refinement: grey lines are projected using the estimate of  $H$  computed by the linear method; white lines are projected using the refined estimate. The white circles are the inlier warped points  $\pi_{k,l}^*$ .

### 3 Visual Odometry (VO): Incremental Updating

The VO module aims essentially at helping in those situations where, knowing an estimate  $H$ , there are not enough known line features visible in the image. To handle this situation, we permanently track a set of locally salient features

in the image, and, when needed, we use frame-to-frame matches to determine  $\Delta H$ , which is the homography from image  $t - 1$  to image  $t$ , i.e.  $\Delta H^t p^{t-1} \sim p^t$  for all points  $p^{t-1}$  in the image  $t - 1$  with counterparts  $p^t$  at time  $t$ .

These points are detected through a Harris detector and tracked by the KLT tracker. One of the main problems here is to modify the Harris detector so that the points are spatially spread over the largest extent possible. For this purpose, we use an adaptive algorithm for point selection [10].



**Fig. 3.** Motion estimation (VO module): tracked points and frame projection into mosaic. Inlier and outlier points are respectively in white and black in the first row. Highlighted rectangular areas are areas where new feature points are searched.

Figure 3 illustrates the process involved in the VO module. It shows motion estimation and the corresponding mosaicking over 150 frames of a short sequence with fast camera rotation. In Fig 3(a) and 3(b), the tracked points are displayed, in white if they correspond to inliers of the  $\Delta H^t$  estimation process, in black otherwise. Note that the number of outliers may be very large in proportion (as in Fig 3(b)), especially under strong blurring situation, so that establishing the correspondences through KLT is meaningful. Figures 3(c) and 3(d) show the corresponding mosaics. Note that whenever the image blur becomes too large, a failure is possible, which is the main limit of this module.



## 4 Using Multiple Homography-Estimation Modules

This section describes the process of combining the homography estimates we obtain from the techniques we described previously.

### 4.1 Estimating the Covariance on Estimated Homographies

A key to the success of any application that seeks to recover geometric information is a proper evaluation of the error associated with the estimated results. We model all the noisy quantities we use (feature positions, estimated objects...) as zero-mean Gaussian random processes. In the computer vision area, there have been two seminal contributions to evaluation and propagation of the covariance matrices, all relying on perturbation theory, i.e. a generic, second-order method [11], and first-order methods that apply to all linear, overdetermined optimization problems solved through matrix spectral decomposition [12][13].

Our procedure is adapted from the aforementioned works, but the nature of the equations is quite different. We propagate uncertainties on points detected in image and uncertainties on model line features (straight lines  $L_i$ ) up to the homography  $H$ . Let us consider that the variance on point  $\pi_{k,l}^*$  is isotropic with value  $\sigma^2$ . The covariance on the parameters  $h$  is denoted as  $\Lambda_h$ . Starting from the Jacobian expressions [13], we get

$$\Lambda_h = J^T \left( \sum_{l=1}^{n_s} \sum_{m=1}^{n_s} (h^T E(\delta s_l^T \delta s_m) h) e_l e_m^T \right) J = J^T \Lambda J \quad (3)$$

where the  $n_s \times 9$  matrix  $J$  is given by  $J_{ij} = -\sum_{l=2}^9 \frac{U_{il} V_{jl}}{\lambda_l}$ ,  $U$ ,  $V$  and  $\{\lambda_l\}$  being the result of the singular value decomposition of  $S$  (left, right singular vectors and corresponding singular values). Vectors  $e_l$  are such that  $e_l(i) = \delta_{li}$ , where  $\delta_{lm}$  is the Kronecker symbol. Assuming noise variances  $\sigma_\alpha^2$  and  $\sigma_c^2$  on the angle, constant line parameters (assumed uncorrelated), and noise variance  $\sigma^2$  on image points coordinates, we finally obtain

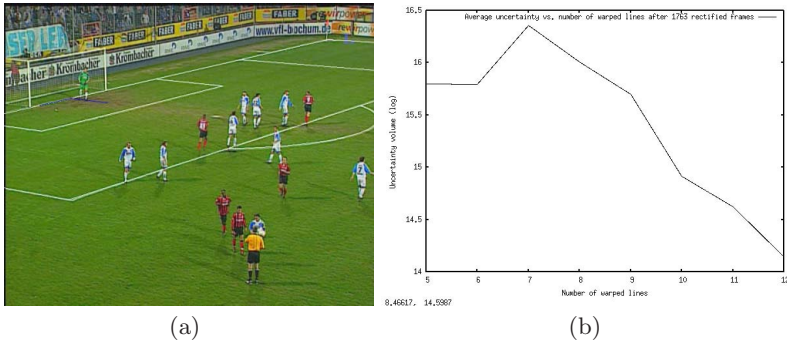
$$\Lambda(i, j) = h^T (\delta_{ij} A_{ij} + \delta_{\iota(i)\iota(j)} B_{ij}) h \quad (4)$$

where  $\iota$  maps row indices in  $S$  onto indices in the set of warped lines. The matrices  $A_{ij}$  and  $B_{ij}$  only depend on the line parameters, point coordinates and their respective uncertainties; they are not detailed here for lack of space.

### 4.2 Propagating Uncertainty Through Rectification Algorithms

Once the covariance  $\Lambda_h$  has been computed, we can use it to evaluate errors on transformed points. As transformed coordinates of a point  $P = (X, Y)$  in the model frame are derived from a multiplication between  $H$  and an image point  $p = (u, v, 1)^T$ , uncertainties on  $P$  naturally combines two terms:

$$\Lambda_P = J_W (H \Lambda_p H^T + J_H \Lambda_h J_H^T) J_W^T \quad (5)$$



**Fig. 4.** Image rectification: (a) projection of the model lines with estimated  $h$  (in white); (b) evolution of the uncertainty volume (determinant of the uncertainty matrix  $\Lambda_P$ ) for the image of a given point with respect to the number of warped lines.

where matrices  $J_W$  and  $J_H$  are Jacobian matrices [12] and matrix  $\Lambda_p$  is the covariance on image points. Typically, this could be the output of a tracking algorithm (uncertainty on the location of the target in the image).

As an illustration, Fig. 5(e) shows the result of error propagation around a homography computed from Fig. 5(a). Red lines in Fig. 5(a) indicate the reprojected model from which the homography has been computed. In Fig. 5(e), at each point from a regular grid, the covariances are represented by a  $3\sigma$  ellipse.

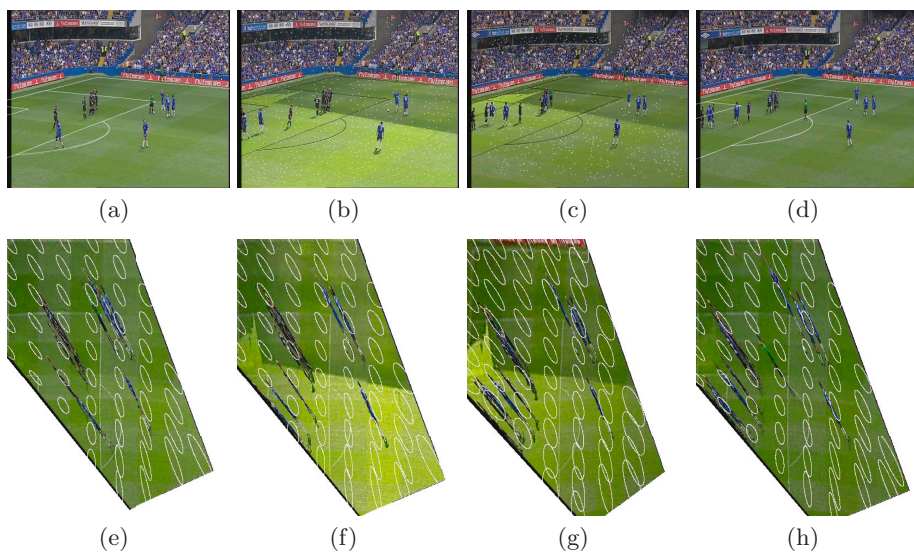
Finally, as expected, the average error decreases with the number of warped lines, as illustrated in Fig. 4(b). A given point was chosen in the image (the center) and the plot shows how a given uncertainty volume in the image domain is transformed in the model domain throughout a given video sequence (where the viewpoint does not change too much). The more warped lines we have, the smaller uncertainties are, on average over several hundreds of frames.

### 4.3 Integrating Uncertain Inter-image Homographies

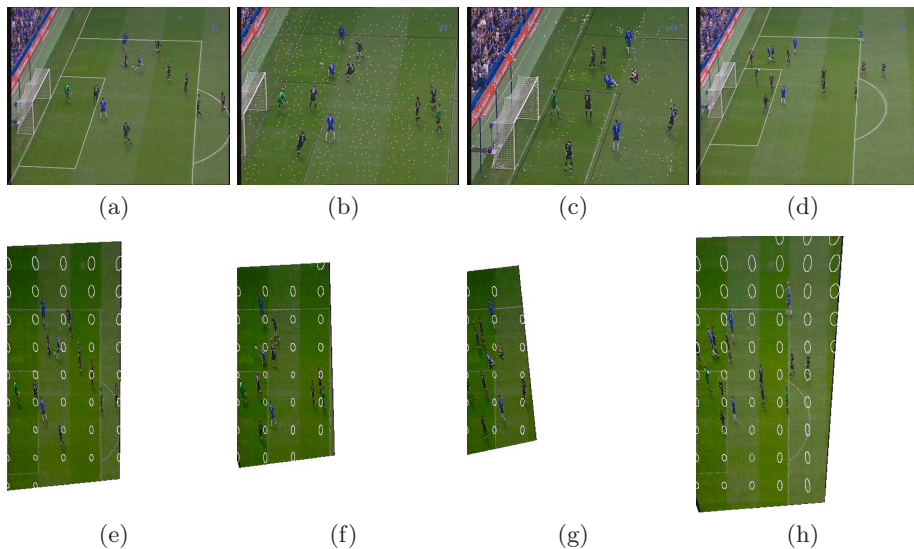
Our policy takes line-based rectification as the first choice. In particular, to save time, inter-image homographies are not computed as long as enough lines are tracked. However, points are detected all over the image at each  $t$ . This allows, when the LT module fails for any reason at time  $t$ , to compute an estimate  $\Delta H^t$  and to update  $H^t$  through  $H^t = H^{t-1} \Delta H^t$ .

In a similar way as in 4.1, uncertainties on  $\Delta H^t$  are computed through Eq. 4 by propagating uncertainties on the matched points, which leads to an estimate of the uncertainty on  $H^t$ . As an illustration, the images of points from the first frame of Fig. 3 (left) into the current frame (right) have accumulated uncertainties that are shown in white in the mosaic image.

Note that when the VO module is used, this uncertainty will tend to grow, as we accumulate errors on relative measures. At some moment, this leads to ambiguities while reprojecting the model (one line taken for another one), so that we switch to the LD module whenever the uncertainty level corresponding to the projection of the center of the image point becomes too large.



**Fig. 5.** Sequence 1 : after 500 frames of line-based rectification (a), the system switches to motion estimation because lighting changes render line warping unsuccessful (b). The inlier points (in white) allow motion estimation for more than 400 frames (b,c), until light conditions become normal and the LT module can restart.

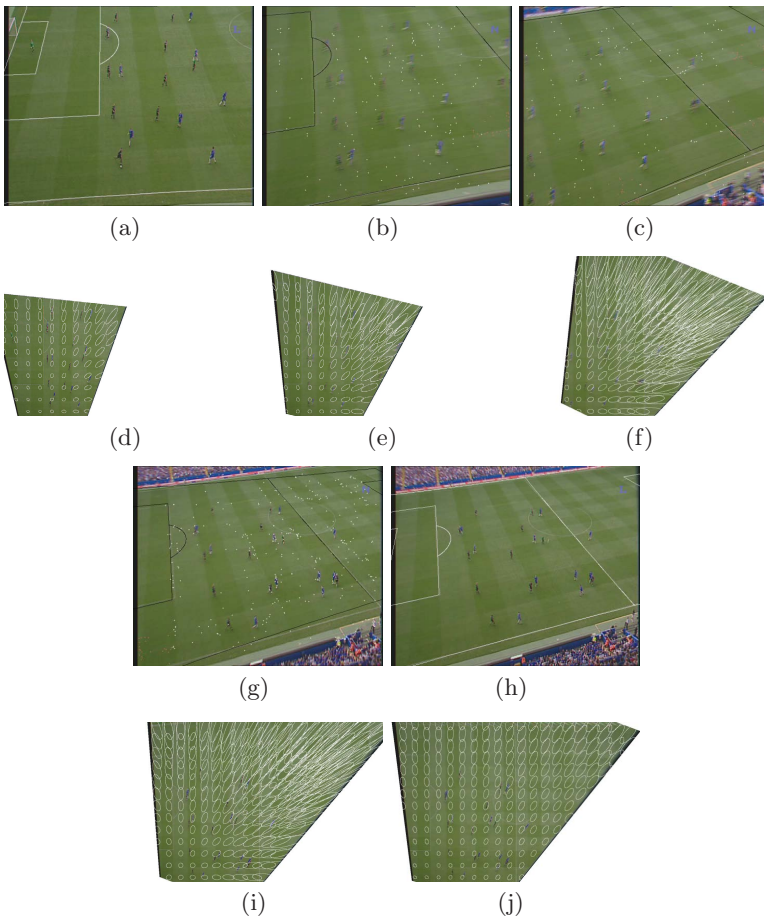


**Fig. 6.** Sequence 2 : line-based rectification is lost at frame 193 (a), motion estimation is done under fast zooming and rotation (b-c), and lines are recovered a frame 300 (d)

## 5 Results

This section presents results of our rectification system for different situations, where the complementarity of the 3 modules we described plays a key role in the success of the continuous, automatic estimation of  $H$ .

A first example in Fig. 5 illustrates the benefits of using a point-based method to estimate increments of homographies, as the VO module (that projects the model onto black lines) helps to maintain an estimate  $H$  while the LT module (that projects the model onto white lines) fails. Indeed, after hundreds of frames rectified by the LT module, sudden, strong lighting changes occur. Starting from frame 529 (Fig. 5(b)), many line segments are strongly reduced in contrast, and



**Fig. 7.** Sequence 3: after staying for a while near the goal area, the camera performs an abrupt motion towards the center, where LT is not possible (c). After unzooming (frame 467, f), goal areas are visible again. Motion estimation has accumulated a lot of uncertainty (e-f-i), but LT is successful on frame 471.

the local warpings are unsuccessful. However, point tracking remains sufficiently reliable for approximately another 400 frames. Even though uncertainties continue to grow during this period, as seen in Figs. 5(f) and 5(g), line tracking resumes without problems in frame 988 (Fig. 5(d)).

Complementarity also appears in Fig. 6 and Fig. 7. These examples show that even under fast motion and strong zooming, when white lines are lost (Fig. 6(b)), it remains possible to rely on motion estimation to recover after a while. However, this motion has to remain local and short, in order to avoid accumulating errors in motion estimation. For example, in Fig. 7, the motion is fast and its amplitude is large and so are the accumulated errors and corresponding uncertainties (see Fig. 7(e), 7(f), 7(i)). This last example is an extreme case : the VO module should have been stopped long before for switching to the LD module but has been let running on purpose to illustrate its limits.

Our implementation runs at about 3Hz on a 3.2GHz Bi-Xeon machine, which is a bit slow, even for  $720 \times 576$  images. Most of the time is spent on point detection and tracking, so our current optimization efforts are focused on it.

## 6 Conclusion and Future Work

We have presented an approach to estimate the image-to-field homography and its uncertainty continuously over sport sequences : after automatic rectification on a single frame, it tracks line features and re-estimates the homography with the constraints induced by image points warped locally onto the corresponding line feature. Finally, it estimates the incremental homography between two frames when line-based rectification is not possible. By way of several challenging examples, we have demonstrated the added benefit of the two techniques.

Our system still needs to be improved to cope with very long sequences. Very fast camera motion causes our system to lose track until sufficient lines come available to allow reinitialization by the first method. Our current work seeks to improve several aspects of the system. First, we would like to perform better filtering of homography estimates in a lower-dimensional parameter space, which implies on-line determination of internal parameters of the camera. Second, as shown in the experimental results, motion estimation through visual odometry accumulates uncertainty and necessarily leads to drift. Currently, we simply switch to LD module when uncertainty becomes too large, but a better answer to this problem would be to identify particularly salient points of the planar scene and incorporate them into the scene model as landmarks.

## References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
2. Ren, J., Orwell, J., Jones, G., Xu, M.: Real-time 3D soccer ball tracking from multiple cameras. In: BMVC 2004. Proc. of the British Machine Vision Conf, pp. 829–838 (2004)

3. Farin, D., Krabbe, S., de With, P., Effelsberg, W.: Robust camera calibration for sport videos using court models. *SPIE Electronic Imaging* 5307 (2004)
4. Hayet, J., Piater, J., Verly, J.: Fast 2D model-to-image registration using vanishing points for sports video analysis. In: *ICIP 2005. Proc. of IEEE Int. Conf. on Image Processing*, Genoa, Italy, vol. 3, pp. 417–420 (2005)
5. Lihe, Q., Luo, Y.: Automatic camera calibration for images of soccer match. In: *Proc. of the Int. Conf. on Computational Intelligence*, pp. 482–485 (2004)
6. Reid, I.D., Zisserman, A.: Goal-directed video metrology. In: Buxton, B.F., Cipolla, R. (eds.) *ECCV 1996. LNCS*, vol. 1065, pp. 647–658. Springer, Heidelberg (1996)
7. Okuma, K., J., L.J., Lowe, D.G.: Automatic rectification of long image sequences. In: Okuma, K. (ed.) *ACCV 2004. Proc. of the Asian Conf. on Computer Vision* (2004)
8. Kim, H., Hong, K.: Robust image mosaicing of soccer videos using self-calibration and line tracking. *Pattern Analysis and Applications* 4, 9–19 (2001)
9. Drummond, T., Cipolla, R.: Real-time tracking of complex structures with on-line camera calibration. In: *BMVC 1999. Proc. of the British Machine Vision Conf*, pp. 574–583 (1999)
10. Brown, M., Szeliski, R., Winder, S.: Multi-image matching using multi-scale oriented patches. In: *CVPR 2005. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 510–517. IEEE Computer Society Press, Los Alamitos (2005)
11. Haralick, R.: Propagating covariance in computer vision. In: *Proc. of ECCV Workshop of Performance Characteristics of Vision Algorithms*, pp. 1–12 (1996)
12. Criminisi, A.: Accurate Visual Metrology from Single and Multiple Uncalibrated Images. PhD thesis, University of Oxford, Dept. Engineering Science (1999)
13. Papadopoulos, T., Lourakis, M.: Estimating the jacobian of the singular value decomposition: Theory and applications. In: Vernon, D. (ed.) *ECCV 2000. LNCS*, vol. 1843, pp. 554–570. Springer, Heidelberg (2000)



# A New Person Tracking Method for Human-Robot Interaction Intended for Mobile Devices<sup>\*</sup>

Rafael Muñoz-Salinas<sup>1</sup>, Eugenio Aguirre<sup>2</sup>, Miguel García-Silvente<sup>2</sup>,  
and Rui Paúl<sup>2</sup>

<sup>1</sup>Department of Computing and Numerical Analysis, University of Córdoba, Spain  
rmsalinas@uco.es

<sup>2</sup>Department of Computer Science and A.I., University of Granada, Spain  
{eaguirre, M.Garcia-Silvente, ruipaul}@decsai.ugr.es

**Abstract.** People detection and tracking are essential capabilities in human-robot interaction. However, the development of these tasks is specially difficult in cluttered environments where it is not possible to create a background model because of the robot movement. To detect and track people in a scene the use of vision sensors is convenient in order to distinguish people from other objects with similar shapes. This paper presents a novel approach for person tracking which combines depth, color and gradient information based on stereo vision. The degree of confidence assigned to depth information in the tracking process varies according to the amount of it found in the disparity map. A novel confidence measure is defined for it. To test the validity of our proposal, it is evaluated in several color-with-depth sequences where people interact in complex situations.

## 1 Introduction

In the last decade, there is a growing interest in the development of systems and techniques for people detection and tracking both in indoor and outdoor environments. The potential applications of that technology has shown to be of great help in several fields such as: ambient intelligent systems [1,2], visual servoing applications [3,4], augmented reality and human-computer interaction [5,6,7,8,9], video compression [10,11] or robotics [12,8,13,14].

People tracking in monocular images is a well explored topic, solved in most of the cases by the integration of multiple visual cues. Nevertheless, it is not a trivial problem, specially when it is not possible to estimate the background of the scene. Unfortunately, it is common when mobile devices are used (e.g. mobile robots). In these cases, tracking based on color histograms is an appropriate method which is able to provide good results at a low computational cost. However,

---

<sup>\*</sup> This work has been partially funded by the Spanish MEC project TIN2006-05565 and the Andalusian Regional Government project TIC1670. *Correspondence to:* Miguel García-Silvente.

pure color-based approaches have a main drawback: they tend to fail when the background color distribution is similar to the target color distribution. In that sense, stereo vision is an interesting sensor that can be employed to provide extra information for enhancing tracking and concentrates now an important interest. The availability of commercial hardware to solve the low-level problems of stereo processing, as well as the lower prices for these devices, turn them into an appealing sensor to be used in intelligent systems. Disparity information is relatively invariable to illumination changes and, therefore, systems that employ stereo vision are expected to be more robust in real scenarios where sudden illumination changes often occur.

This paper presents a solution to the person tracking problem by the integration of multiple visual cues using a particle filtering approach [15]. They are specially interesting because they are able to deal naturally with systems where both the posterior density and the observation density are non-Gaussian.

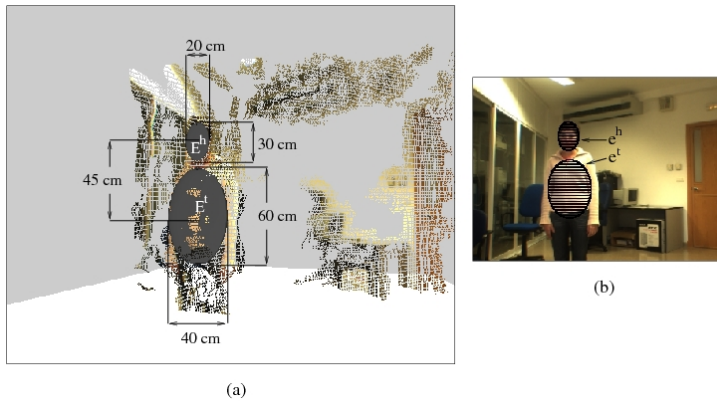
People detection and tracking are interesting topics that have attracted the interest of researches in many areas. These problems have been generally tackled by exploiting the morphological characteristics of the human body [16] and the color characteristics of human skin [17,18,19]. Because of static color models usually leads to a degradation of the performance [20] (due to illumination changes), some authors have opted to use dynamic skin color models [21]. Others, have provided extra information to the tracker by employing additional information such as the color of the clothes of the user in order to cope with the illumination problem and to avoid confusion between users [22,23,24]. Other ways of dealing with the detection problem include the use of multiple sensors [25], multiple visual cues [12] or specific hardware [26] to perform a more robust detection.

Several authors have employed stereo vision for developing more sophisticated tracking methods taking advantage of stereo vision. In the vast majority of the cases they employ a background model of the scene in order to ease the process. However, background modelling is not possible in many applications, e.g., when the stereo system must be on a mobile device such as a mobile robot. Besides, disparity calculation is not always possible because of occlusions or absence of texture. Nevertheless, most of the previously mentioned works do not deal explicitly with this problem. In this paper we propose a method based on particle filters which is able to deal with the above problems.

## 2 Proposed Method

This section explains our person tracking method. It is based on the use of the 3D body model shown in Fig. 1(a). It is a model comprised by two planar ellipses and the information projected inside them: one fitting the head region of the person ( $E^h$ ), and another one fitting their torso ( $E^t$ ). In an initial phase, the model must be appropriately placed to fit the person's head and torso (e.g. using a face detector [27]). Then, two color models (one for each ellipse) are stored to be employed in order to track the person. The *HSV* color space has been employed in this work because it is relatively invariable to illumination changes.





**Fig. 1.** (a) Anatomical measures used for the different human body sections, represented over real stereo information corresponding to a scene appearing a person. (b) Projection of the human model on the reference camera image (shown a person) .

Our tracking approach employs the Condensation algorithm where particles represent positions and velocities of the 3D model. The 3D position of the model is given by the central position of its upper ellipse  $E_c^h = (X, Y, Z)$  that corresponds to the person's head being tracked. Given a 3D position for the model, it is possible to determine its projection on the reference camera image. Figure 1(b) shows the projection of the 3D model shown in Fig. 1(a). Particles weights are calculated by first projecting the 3D model on the reference camera image and then examining the inner pixels of the projected ellipses. If a particle is near the true person's location, then the inner pixels of the model projection must have a color distribution similar to the target color models and be at the distance indicated by the particle. Besides, the gradient around the upper ellipse should indicate the presence of an elliptical object (person's head). Nevertheless, these assumptions are very strict and several contingencies must be taken into account. First, as previously mentioned, the disparity calculation is subject to errors. Second, in some cases it might be impossible to determine the disparity of the target region because of occlusions or absence of texture. As follows, we explain in detail how color and depth information have been combined in our approach to deal with these problems.

## 2.1 Initial Phase and Model Projection

The 3D model employed is comprised by two ellipses whose sizes have been selected according to standard people sizes. The ellipses axis lengths are shown in Fig. 1(a). Let be  $E_w^h$  and  $E_h^h$  the horizontal and vertical lengths of the axis of  $E^h$ . As it can be seen, the axis of  $E^t$  are twice longer than the axis of  $E^h$ .

In the initial phase, the model must be appropriately placed to fit the head and torso of the person in order to create the two target color models. They are employed in the tracking phase in order to look for similar colored regions in

the subsequent images. The first target color model, named  $\hat{q}_{E^t}$ , corresponds to the torso of the person being tracked and stores information about the person’s clothes. The second color model,  $\hat{q}_{E^h}$ , corresponds to the person’s head region.

The two ellipsoidal surfaces of the model ( $E^h$  and  $E^t$ ) project in two ellipses on the reference camera image (let us denote them by  $e^h$  and  $e^t$ ). The centre of  $e^h$  (let us denote it  $e_c^h = (e_x^h, e_y^h)$ ) is the projection of  $E_c^h = (X, Y, Z)$ , that can be calculated using projective geometry as:

$$e_x^h = \frac{Xf}{Z}; e_y^h = -\frac{Yf}{Z}. \tag{1}$$

The sizes of the horizontal and vertical axis of  $e^h$ , let us denote them  $e_w^h$  and  $e_h^h$ , can be calculated using projective geometry as:

$$e_w^h = \frac{ZE_w^h}{f}; e_h^h = \frac{ZE_h^h}{f}. \tag{2}$$

The projection of  $E^t$  can be calculated using the same procedure, obtaining  $e^t$ . The color models of the torso and head projected ellipses (let us denote them by  $\hat{q}_E^t$  and  $\hat{q}_E^h$  respectively) are stored in order to look for the person in the tracking phase.

### 2.2 Tracking Phase

Let a particle  $s_i(t) = [X_i(t), Y_i(t), Z_i(t), \dot{X}(t), \dot{Y}(t), \dot{Z}(t)]$  represents the position and speed of the person being tracked. The sample set is propagated using a dynamic model

$$s(t) = As(t - 1) + w(t - 1), \tag{3}$$

where  $A$  indicates the deterministic component of the model and  $w(t - 1)$  is a multivariate Gaussian random variable. We have opted for a first order model where  $A$  describes the target moving at constant velocity  $(\dot{X}(t), \dot{Y}(t), \dot{Z}(t))$ .

As previously indicated, for each particle  $s_i(t)$ , it is calculated its projection on the reference camera image. Each particle projects as two ellipses  $e_i^h(t)$  and  $e_i^t(t)$ . Our approach consists in examining color and depth of the projected ellipses and the gradient information around the upper one. For the sake of clarity, it is explained first how color and depth information are modelled for each projected ellipse, and then it is explained how gradient information is examined for the upper one.

For our approximation both color and depth information are considered as “normal-behaved” because the distribution of the information is expected to be more similar to its neighbourhood than to further information.

Colour information is managed by defining the variable  $d_i^h(t) \sim N(0, \sigma_c)$  that is the Bhattacharyya distance:

$$d_i^h(t) = \sqrt{1 - \rho(\hat{q}_E^h, \hat{q}_{e,i}^h(t))}. \tag{4}$$

It provides values near 0 when two color models are similar and tends to 1 as they differ. In Eq. 4,  $\hat{q}_{e,i}^h(t)$  is the color model of  $e_i^h(t)$  and  $\hat{q}_E^h$  is the target color model of the person's head.

Depth information is managed by the variable  $\mu_{z,i}^h(t) \sim N(Z_i(t), \sigma_z)$  that is defined as:

$$\mu_{z,i}^h(t) = K \sum_{j=1}^n w \left( \frac{\|e_{c,i}^h(t) - p_{j,i}^h(t)\|}{a} \right) I_z(p_{j,i}^h(t)). \tag{5}$$

where  $I_z$  represents the *distance image* obtained from the disparity map, each pixel  $I_z(p)$  represents the  $Z$  component of the point  $p$ ,  $w$  is the weighting function defined as:  $w(r) = \begin{cases} 1 - r^2 & \text{if } r < 1 \\ 0 & \text{otherwise} \end{cases}$ ,  $a$  is the distance from the farthest point of the ellipse to its centre  $e_{c,i}^h(t)$ ,  $K$  is a normalisation constant calculated by imposing the condition that  $\sum_{j=1}^n w \left( \frac{\|e_{c,i}^h(t) - p_{j,i}^h(t)\|}{a} \right) = 1$  and  $\{p_{j,i}^h(t)\}_{j=1..n_i(t)}$  are the inner pixels of  $e_i^h(t)$ . The variable  $\mu_{z,i}^h(t)$  represents the average distance of the pixels enclosed in  $e_i^h(t)$ , assigning more relevance to central pixels (using  $w$ ). Assigning more relevance to central pixels helps to reduce the influence of occluding objects in the target boundaries.

It must be reminded that  $I_z$  might contain undefined values (unmatched points) so that Eq. 5 is only applied for these pixels  $p_{j,i}^h(t)$  whose distance is known. Thus, the value provided by  $\mu_{z,i}^h(t)$  is affected by uncertainty since there might be unmatched points that if detected might alter its value. Our intention is to manage the possible absence of depth information into the model in order to do it more robust. The greater the amount of disparity found, the higher the degree of confidence assigned to depth information is. The problem is then to define a probability distribution function that merges the original distribution taking into account the degree of confidence in  $\mu_{z,i}^h(t)$ . Our proposal consists in calculating a confidence measure that is included in the standard deviation of the probability distribution function of  $\mu_{z,i}^h(t)$ . The idea is to modify the shape of the normal distribution so that when the confidence in depth information is high, the new distribution is exactly like the original one. However, as the confidence on depth information decreases, the standard deviation of the probability distribution function is increased making the distribution more similar to an uniform one.

Lets us denote as  $\lambda^h(t)$  the confidence measure that indicates the proportion of valid points detected in the inner pixels of all the upper projected ellipses  $(e_i^h(t)_{i=1..N})$  respect to the total points analysed:

$$\lambda^h(t) = \frac{\sum_{i=1}^N \sum_{j=1}^{n_i} \delta(p_{j,i}^h(t))}{\sum_{i=1}^N n_i(t)}. \tag{6}$$

In Eq. 6,  $N$  is the number of particles and  $\delta$  is a function that only has two values: it is 0 when the pixel  $p_{j,i}^h(t)$  has an undefined distance value and 1 in the

opposite case. Thus, the value  $\lambda^h(t)$  is in the range  $[0, 1]$ , where 1 means that for each particle all the pixels in all the projected ellipses have a known distance value, and decreases to 0 as the number of unmatched points increases.

Using the above calculated  $\lambda^h(t)$ , the probability distribution of depth information is redefined for  $\lambda^h(t) \neq 0$  as:

$$\mu_{z,i}^h(t) \sim N(Z_i(t), \sigma_z^h(t)),$$

where

$$\sigma_z^h(t) = \frac{\sigma_z}{\lambda^h(t)}.$$

The probability distribution goes from a normal with the mass of the probability around  $Z_i(t)$  to, little by little, a distribution with all the values with the same probability. So, when  $\lambda^h(t) = 0$ ,  $\mu_{z,i}^h(t)$  follows an uniform distribution.

We define the joint probability distribution function of color and depth for the upper ellipse, when  $\lambda^h(t) \neq 0$ , as:

$$P_{cd}(e_i^h(t)) = \frac{1}{2\pi\sigma_c\sigma_z(t)} \exp\left(-\frac{1}{2}\left(\frac{d_i^h(t)^2}{\sigma_c^2} + \frac{(\mu_{z,i}^h(t) - Z_i(t))^2}{\sigma_z(t)^2}\right)\right) \quad (7)$$

When  $\lambda^h(t) = 0$ ,  $\mu_{z,i}^h(t)$  is an uniform and any value is equally probable. In case of total absence of disparity ( $\lambda^h(t) = 0$ ), our approach performs as pure color-based tracker.

For the torso ellipse  $e_i^t(t)$ , we proceed in a similar way in order to define the probability distribution function  $P_{cd}(e_i^t(t))$ .

Finally, we also aim to detect whether the projected ellipse perimeter  $e_i^h(t)$  is placed on an ellipsoidal object by analyzing the image gradient. This is a technique employed by several authors in the related literature [28,29,30]. We have opted for using a variant of the Birchfield’s method [29] that evaluates the gradient direction of the ellipse perimeter. We define the measure  $fitting_i(t)$  as:

$$fitting_i(t) = 1 - \frac{1}{N} \sum_{j=1}^N |n_j \cdot g_j|, \quad (8)$$

where  $N$  is the total number of pixels in the perimeter of the ellipse  $e_i^h(t)$ ,  $(\cdot)$  denotes the dot product,  $g_j$  is the unit gradient vector of the image at the  $j$ -th pixel of the perimeter and  $u_j$  is the unit vector normal to the ellipse at pixel  $j$ . Assuming  $fitting_i(t) \sim N(0, \sigma_g)$ , its probability distribution function is defined as:

$$\phi_i(t) = \frac{1}{\sqrt{2\pi}\sigma_g} \exp\left(-\frac{fitting_i(t)^2}{2\sigma_g^2}\right) \quad (9)$$

Using the distributions explained above, and assuming independence between them, the final weight of a particle is calculated as:

$$\pi_i(t) = P_{cd}(e_i^h(t))P_{cd}(e_i^t(t))\phi_i(t) \quad (10)$$

Equation 10 is able to manage uncertainty in the depth information. In the worst case (absence of information about disparity), the weight of a particle is based on color and gradient information uniquely. However, the greater the amount of disparity found, the greater its influence on the final particle weight. The final person position is assumed to be the mean of the state  $\mathcal{E}[S(t)]$ .

Assuming independence in Eq. 10 allows us to speed up the particle computation. For each particle, the value  $P_{cd}(e_i^h(t))$  is calculated first. If it has a low value, the final particle weight will also be low. Thus, we can save up computing time by avoiding the calculation of  $P_{cd}(e_i^t(t))$  and  $\phi_i(t)$  when  $P_{cd}(e_i^h(t))$  is sufficiently low.

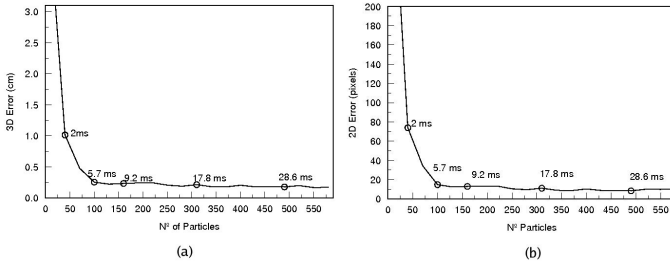
Finally, the target color models  $\hat{q}_{E^h}$  and  $\hat{q}_{E^t}$  are updated at the end of each iteration step in order to adapt the tracking process to illumination changes. However, the target color models are only updated when the weight of the final estimated state  $\pi_{\mathcal{E}[S]}$  is above a certain threshold  $\pi_T$  in order to avoid including as part of the updated models elements from the background or from occluding objects. The target color models are updated as proposed by [31] using the projection of the 3D model indicated by  $\mathcal{E}[S(t)]$ .

### 3 Experimental Results

This section explains the experimentation carried out in order to validate our proposal. For experimentation purposes, several color-with-depth sequences have been recorded using a *Bumblebee* stereo camera from the *Point Grey Research* manufacturer. The stereo system is comprised by two coplanar cameras separated by a distance of 12 cm and is able to record sequences of size  $320 \times 240$  at a frame rate of 15 fps. The stereo correspondence algorithm employed is an improved version of SAD (provided by the manufacturer) that performs sub-pixel interpolation. The recorded sequences show scenes with a varying number of people (from one up to four) interacting in a room. In the sequences, people perform several types of interactions: walk at different distances, shake hands, cross their paths, jump, run, embrace each other and even quickly swap their positions trying to confuse the system. People were instructed not to walk farther than 6 m from the camera. At larger distances the depth errors obtained became too high because of the narrow baseline of the stereo system. A total of 7 different people participated in the tests.

Our experimentation aims to evaluate three aspects of the method proposed. First, the tracking error in determining the 3D position of the person being tracked. Second, the tracking error in determining the 2D person's head position in the reference camera image. Third, the computing time of our method. In order to obtain quantitative measures of the tracking error, the people's head position have been manually determined in each frame of the sequences. In total, there have been manually extracted 4460 positions from the sequences recorded.

In unimodal problems such as this, the final mean state  $\mathcal{E}[S(t)]$  might be considered as the best person's position estimation. Thus, the 2D tracking error is calculated as the distance from the manually determined position to the upper



**Fig. 2.** (a) Tracking error in determining the 3D person position (b) Tracking error in determining the 2D head position

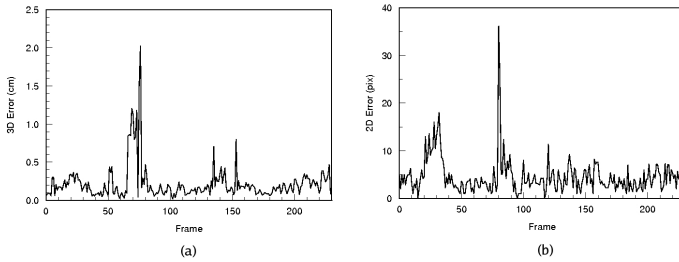
ellipse center when the 3D model is projected from  $\mathcal{E}[S(t)]$ . For determining the 3D tracking error, a small region around the manually determined head position has been selected. Then, the mean 3D position of the region has been assumed to be the 3D person’s position in the room. Of course, it is not the real 3D person position but an estimation subject to stereo errors. The 3D tracking error in each frame has been computed as the distance from the 3D position estimation to  $\mathcal{E}[S(t)]$ .

As previously indicated, the performance of methods based on particle filters increases as the number of particles grows. However, the higher the number of particles employed, the higher the computational time required for the algorithm is. Therefore, it is important to analyze the error of the tracker as a function of the number of particles in order to decide the most appropriate configuration for a particular application. Therefore, each sequence has been evaluated for an increasing number of particles. However, because of the stochastic nature of the algorithm, each test has been repeated several times with different seeds for the random number generator. In order to run the tests, the algorithm parameters have been experimentally determined as  $\sigma_z = 0.1$ ,  $\sigma_c = 0.2$  and  $\sigma_g = 0.3$ .

The analysis of Fig. 2(a-b) reveals that for a low number of particles, the algorithm obtains relatively high errors. However, there is a rapid improvement of the performance as the number of particles grow up to the limit of 100 particles. As it can be noticed, no relevant improvements are achieved above this limit. The average computing time per iteration required for 100 particles is 5.7 ms. Thus, we can consider that the proposed method is valid for real-time tracking purposes. Following, we show the tracking results from three of the sequences employed in our tests. All the sequences employed to test our algorithm, showing the tracking results for  $N = 100$  particles, are publicly available at <http://decsai.ugr.es/isg/salinas/stpfpeopletracking/>

The evolution of the tracking errors along the frames of a test sequence are shown in Fig. 3(a-b).

The graphs in Fig. 3 represent, in the horizontal axis, the frame numbers. In Fig. 3(a), the vertical axis represents the tracking error in determining the 3D person’s position and in Fig. 3(b), the vertical axis represents the 2D tracking error. It can be observed in Fig. 3(a-b) that the error is restricted to small values and that the highest peak occurs around frame #110, when the first



**Fig. 3.** Tracking error of a test sequence: (a) in 3D position, (b) in projection 2D

total occlusion takes place. Let us denote by  $\mu_{3De}$  the average 3D tracking error of a complete sequence, and by  $\sigma_{3De}$  its standard deviation. In that sequence, the average 3D tracking error is  $\mu_{3De} = 0.139$  m with a standard deviation  $\sigma_{3De} = 0.091$  m. Let us also denote by  $\mu_{2De}$  and by  $\sigma_{2De}$  the average error and standard deviation of the 2D tracking error in the sequence. For that particular case, we obtained  $\mu_{2De} = 4.4$  pix and  $\sigma_{2De} = 3.6$  pix. Please notice that the tracker is able to fit appropriately the upper ellipse to the woman's head while she performs a steady movement. Although when she performs fast movements the head position estimation has a higher error (frames #120–143), it is immediately corrected when the movement is reduced.

## 4 Conclusions

An approach to the person tracking problem based on combining multiple visual cues using a particle filtering approach is presented. However, our method employs a 3D rigid human body model comprised by two ellipses: one for tracking the person's head and another one for his/her the torso. Particles represent possible 3D positions for the model that are evaluated by examining their projection in the camera image. Our method integrates depth, color and gradient information to perform a robust tracking without creating a background model of the environment. Thus, it is an appropriate method for mobile systems.

Depth information cannot be always extracted because of occlusions or absence of texture. Our method is able to deal with this problem by defining a certainty measure that indicates the degree of confidence in depth information. The confidence measure is employed to modify the probability distribution function employed for weighting the particles. The greater is the amount of disparity found, the greater is its contribution to the final particles weights and vice versa. In the worst case (absence of information about disparity), the proposed algorithm makes use of the information available (color and gradient) to perform the tracking. The proposed algorithm does not only determine the 3D person position but also his/her head position in the camera image. This is a very valuable piece of information for human-computer and human-robot interaction tasks (e.g., face pose estimation, expression analysis).

Several color-with-depth sequences have been employed in order to test the validity of our proposal. The sequences recorded show a varying number of people

(from one up to four) interacting in a room. In the sequences, people perform different types of interactions: walk at different distances, shake hands, cross their paths, jump, run, embrace each other and even quickly swap their positions trying to confuse the system. The tracking errors have been calculated for different number of particles in order to determine the number of them that allows an appropriate trade-off between tracking error and computing time. The experimental results show that the proposed method is able to determine, in real-time, both the 3D position and the 2D head position in the camera image of a moving person despite of the presence of other people. Besides, the proposed method is able to deal with both partial and short-term total occlusion.

## References

1. Hayashi, K., Hashimoto, M., Sumi, K., Sasakawa, K.: Multiple-person tracker with a fixed slanting stereo camera. In: 6th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 681–686. IEEE Computer Society Press, Los Alamitos (2004)
2. Patil, R., Rybski, P., Kanade, T., Veloso, M.: People Detection and Tracking in High Resolution Panoramic Video Mosaic. In: IROS 2004. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1323–1328 (2004)
3. Argyrs, A., Lourakis, M.: Three-dimensional tracking of multiple skin-colored regions by a moving stereoscopic system. *Applied Optics* 43, 366–378 (2004)
4. Malis, E., Chaumette, F., Boudet, S.: 2 1/2d visual servoing. *IEEE Transactions on Robotics and Automation* 15, 238–250 (1999)
5. Colombo, C., Bimbo, A., Valli, A.: Visual Capture and Understanding of Hand Pointing Actions in a 3-D Environment. *IEEE Transactions On Systems, Man and Cybernetics - Part B* 33, 677–686 (2003)
6. Darrell, T., Gordon, G., Harville, M., Woodfill, J.: Integrated Person Tracking Using Stereo, Color, and Pattern Detection. *Int. Journ. Computer Vision* 37, 175–185 (2000)
7. Grest, D., Koch, R.: Realtime multi-camera person tracking for immersive environments. In: IEEE 6th Workshop on Multimedia Signal Processing, pp. 387–390. IEEE Computer Society Press, Los Alamitos (2004)
8. Kahn, R., Swain, M., Prokopowicz, P., Firby, R.: Gesture recognition using the perseus architecture. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 734–741. IEEE Computer Society Press, Los Alamitos (1996)
9. Wren, C., Azarbayejani, A., Darrell, T., Trevor, P.A.: Pfnder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 780–785 (1997)
10. Menser, B., Brunig, M.: Face detection and tracking for video coding applications. In: Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, pp. 49–53 (2000)
11. Vieux, W.E., Schwerdt, K., Crowley, J.: Face-Tracking and Coding for Video Compression. In: Int. Conf. Computer Vision Systems, pp. 151–160 (1999)
12. Böhme, H., Wilhelm, T., Key, J., Schauer, C., Schröter, C., Hempel, H.G.T.: An approach to multi-modal human-machine interaction for intelligent service robots. *Robotics and Autonomous Systems* 44, 83–96 (2003)



13. Muñoz-Salinas, R.M., Aguirre, E., García-Silvente, M., González, A.: A fuzzy system for visual detection of interest in human-robot interaction. In: ACIDCA-ICMI 2005. 2nd International Conference on Machine Intelligence, pp. 574–581 (2005)
14. Muñoz-Salinas, R.M., Aguirre, E., García-Silvente, M., González, A.: People detection and tracking through stereo vision for human-robot interaction. *Lectures Notes on Artificial Intelligence*, pp. 337–346 (2005)
15. Isard, M., Blake, A.: CONDENSATION – conditional density propagation for visual tracking. *Int. J. Computer Vision* 29, 5–28 (1998)
16. Hirai, N., Mizoguchi, H.: Visual tracking of human back and shoulder for person following robot. In: IEEE/ASME. International Conference on Advanced Intelligent Mechatronics, vol. 1, pp. 527–532 (2003)
17. Ghidary, S., Nakata, Y., Takamori, T., Hattori, M.: Human detection and localization at indoor environment by home robot. *IEEE International Conference on Systems, Man, and Cybernetics* 2, 1360–1365 (2000)
18. Saito, H., Ishimura, K., Hattori, M., Takamori, T.: Multi-modal human robot interaction for map generation. In: SICE 2002. 41st SICE Annual Conference, vol. 5, pp. 2721–2724 (2002)
19. Sidenbladh, H., Kragic, D., Christensen, H.: A Person Following Behaviour for a Mobile Robot. In: IEEE International Conference on Robotics and Automation, vol. 1, pp. 670–675 (1999)
20. Martinkauppi, B., Soriano, M., Pietikainen, M.: Detection of skin color under changing illumination: a comparative study. In: 12th International Conference on Image Analysis and Processing, pp. 652–657 (2003)
21. Sigal, L., Sclaroff, S., Athitsos, V.: Skin color-based video segmentation under time-varying illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 862–877 (2004)
22. Kwolek, B.: Color vision based person following with a mobile robot. In: Third International Workshop on Robot Motion and Control, vol. 1, pp. 375–380 (2002)
23. Muñoz-Salinas, R., Aguirre, E., García-Silvente, M.: People detection and tracking using stereo vision and color. *Image and Vision Computing* 25, 995–1007 (2007)
24. Schlegel, C., Illmann, J., Jaberg, H., Schuster, M., Worz, R.: Vision based person tracking with a mobile robot. In: BMVC 1998. 9th British Machine Vision Conference, vol. 1, pp. 418–427 (1998)
25. Fritsch, J., Kleinhagenbrock, M., Lang, S., Plötz, T., Fink, G.A., Sagerer, G.: Multi-modal anchoring for human-robot interaction. *Robotics and Autonomous Systems* 43, 133–147 (2003)
26. Wilhelm, T., Böhme, H., Gross, H.: A multi-modal system for tracking and analyzing faces on a mobile robot. *Robotics and Autonomous Systems* 48, 31–40 (2004)
27. Yang, M., Kriegman, D., Ahuja, N.: Detecting faces in images: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24, 34–58 (2002)
28. Baumberg, A., Hogg, D.: An efficient method for contour tracking using active shape models. In: IEEE Workshop on Motion of Non-Rigid and Articulated Objects, pp. 194–199. IEEE Computer Society Press, Los Alamitos (1994)
29. Birchfield, S.: Elliptical Head Tracking Using Intensity Gradients and Color Histograms. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 232–237. IEEE Computer Society Press, Los Alamitos (1998)
30. Blake, A., Curwen, R., Zisserman, A.: A framework for spatiotemporal control in the tracking of visual contours. *Int. Journal of Computer Vision* 11, 127–145 (1993)
31. Nummiaro, K., Koller-Meier, E., Gool, L.: An Adaptive Color-Based Particle Filter. *Image and Vision Computing* 21, 99–110 (2003)

# Example-Based Face Shape Recovery Using the Zenith Angle of the Surface Normal

Mario Castelán<sup>1</sup>, Ana J. Almazán-Delfín<sup>2</sup>, Marco I. Ramírez-Sosa-Morán<sup>3</sup>,  
and Luz A. Torres-Méndez<sup>1</sup>

<sup>1</sup> CINVESTAV Campus Saltillo, Ramos Arizpe 25900, Coahuila, México  
[mario.castelan@cinvestav.edu.mx](mailto:mario.castelan@cinvestav.edu.mx)

<sup>2</sup> Universidad Veracruzana, Facultad de Física e Inteligencia Artificial, Xalapa 91000,  
Veracruz, México

<sup>3</sup> ITESM, Campus Saltillo, Saltillo 25270, Coahuila, México

**Abstract.** We present a method for recovering facial shape using an image of a face and a reference model. The zenith angle of the surface normal is recovered directly from the intensities of the image. The azimuth angle of the reference model is then combined with the calculated zenith angle in order to get a new field of surface normals. After integration of the needle map, the recovered surface has the effect of mapped facial features over the reference model. Experiments demonstrate that for the lambertian case, surface recovery is achieved with high accuracy. For non-Lambertian cases, experiments suggest potential for face recognition applications.

## 1 Introduction

Acquiring surface models of faces is an important problem in computer vision and visualization, since it has significant applications in biometrics, computer games and production graphics. Shape-from-shading (SFS) [1] seems to be an appealing method, since this is a non-invasive process which mimics the capabilities of the human vision system [2]. For face shape recovery, however, the use of SFS has proved to be an elusive task, since the concave-convex ambiguity can result in the inversion of important features such as the nose. To overcome this problem, domain specific constraints have proved to be essential to improve the quality of the overall reconstructions, and the recovery of accurately detailed facial surfaces still proves to be a challenge.

Despite the improvements achieved by using domain specific information, it is fair to say that no SFS scheme has been demonstrated to work as statistical SFS [3,4]. In this framework, the main idea is to represent surfaces in the parametric eigenspace. This is constructed through the eigenvectors of the covariance matrix of a training set of 3D faces. Once the surfaces are parameterized, shape-coefficients that satisfy image irradiance constraints are sought. Unfortunately, a computationally expensive parameter search has to be carried out, since the fitting procedure involves minimizing the error between the rendered facial surface and the observed intensity of the input image. This minimization procedure is subject to multiple local minima.

More recently, Hancock et al. [5,6] have tried to relax this problem by using different surface representations and alternative parameter fitting procedures. They have proposed statistical models that can be fitted to image brightness data using geometric constraints on surface normal direction provided by Lambert's law [7].

Kemelmacher and Basri [8] have developed a novel method for 3D facial shape recovery from a single image using a single 3D reference surface height model of a different face. This example-based technique "molds" the reference model to the input image to achieve surface reconstruction. Their method seeks the shape, albedo, and lighting that best fit the image, while preserving the overall structure of the model. Although this method does not use statistical models of faces, good results can be achieved provided that the reference model shows a good resemblance to the input image.

In this paper we test the simple idea of using the zenith angle of the surface normal to "map" facial features from an input image to a reference model. The zenith angle is calculated directly from the image intensities. This information is further coupled with the azimuth angle of a reference model in order to obtain a set of surface normals. The final result, however, is achieved only until these surface normals are integrated. Experiments over Lambertian data (i.e. ideal data) show that facial shape can be accurately recovered through the combination of zenith and azimuth angles. In a similar way, experiments with non-Lambertian data suggest a potential for face recognition applications.

The paper is organized as follows: in Section 2 we explain concepts related to surface orientation, Section 3 describes the image irradiance equation. The combination of zenith and azimuth angles to recover facial surfaces is explained in Section 4. An experimental evaluation of the model is described in Section 5. We finally present conclusions and future work in Section 6.

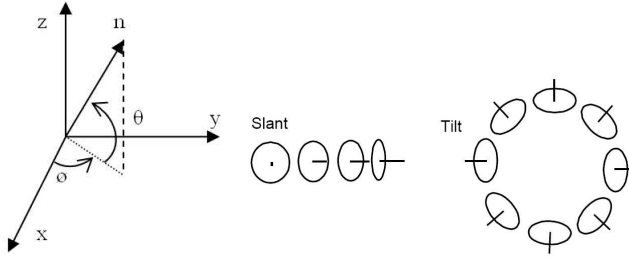
## 2 Surface Orientation

Information about a surface that is intermediate between a full 3D representation and a 2D projection onto a plane is often referred to as a 2.5D surface representation [9]. Surface orientation is one of the most important 2.5D representations. For every visible point on a surface, there exists a corresponding orientation which is usually represented by either surface normal, surface gradient or the azimuth and zenith angles of the surface normal.

In contrast to height data, directional information cannot be used to generate novel views in a straightforward way. However, given the illumination direction and the surface albedo properties, then the direction of the surface normal plays a central role in the radiance generation process. This is of particular interest in face analysis since light-source effects are responsible for more variability in the appearance of facial images than changes in shape or identity [10].

The *surface gradient* is based on the directional partial derivatives of the height function  $Z$ ,

$$p = \frac{\partial Z(x, y)}{\partial x} \quad \text{and} \quad q = \frac{\partial Z(x, y)}{\partial y}. \quad (1)$$



**Fig. 1.** The azimuth ( $\phi$ ) and zenith ( $\theta$ ) angles of a surface normal (left) and the visual interpretation of the slant and the tilt (right)

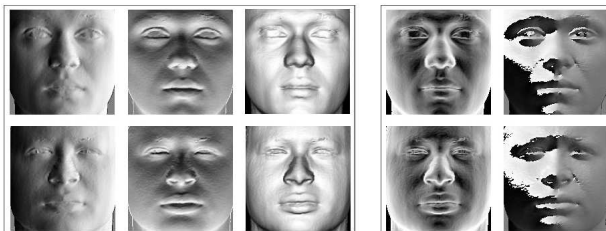
The set of first partial derivatives of a surface is also known as the gradient space. This is a 2D representation of the orientation of visible points on the surface.

The *surface normal* is a vector perpendicular to the plane tangent to a point of the surface. The relation between surface normal and surface gradient is given by

$$(n_x, n_y, n_z) = \frac{(p, q, -1)}{\sqrt{p^2 + q^2 + 1}}. \tag{2}$$

Directional information can also be expressed using the zenith and azimuth angles of the surface normals. In terms of the slope parameters, the zenith angle is  $\theta = \arctan \sqrt{p^2 + q^2}$  and the azimuth angle is  $\phi = \arctan \frac{q}{p}$ . Here we use the four quadrant arc-tangent function and therefore  $-\pi \leq \phi \leq \pi$  and  $0 \leq \theta \leq \pi/2$ . The zenith angle is related to inclination and the azimuth angle is related to orientation. We will refer to these two angles as slant and tilt (see Figure 1).

Figure 2 presents examples of facial surfaces. The figure is divided into two panels. The leftmost panel illustrates surface normals:  $n_x$ ,  $n_y$  and  $n_z$  appear from left to right. The slant and tilt related to these surface normals are shown, respectively, in the columns of the rightmost panel. Two different examples are



**Fig. 2.** The two rows of the figure present two different subjects. The three columns of the leftmost panel show  $n_x$ ,  $n_y$ ,  $n_z$ . Slant and tilt are shown in the two columns of the rightmost panel.

shown row-wise. We present images as intensity maps, where brighter and darker pixels correspond to higher and lower values for each measure. Note how the surface normals seem to characterize a face illuminated from three orthogonal directions. From the slant, we can observe which surface normals have a steep inclination (darker intensities) and which have a small inclination (brighter intensities). High slant values are located around regions such as forehead, tip of the nose, chin, and centers of the eyes and mouth. Facial boundaries and sides of the nose correspond to low slant values. Another important feature to note from the figure is the similarity between  $n_z$  and the slant. This similarity will be discussed in the next section.

### 3 The Image Irradiance Equation

The shape-from-shading (SFS) problem is the one of recovering the surface that, after interaction with the environment (illumination conditions, objects' reflectance properties, inter-reflections), produces the radiances perceived by human eyes as intensities.

In brief, SFS aims to solve the image irradiance equation,  $E(x, y) = R(p, q, \mathbf{s})$ , where  $E$  is the image brightness value of the pixel with position  $(x, y)$ , and  $R$  is a function referred to as *the reflectance map* [11]. The reflectance map uses the surface gradients  $p = \frac{\partial Z(x, y)}{\partial x}$  and  $q = \frac{\partial Z(x, y)}{\partial y}$  together with the light source direction vector  $\mathbf{s}$  to compute a brightness estimate which can be compared with the observed brightness.

If the surface normal at the image location  $(x, y)$  is  $\mathbf{n}(x, y)$ , then under the Lambertian reflectance model, with a single light source direction, no inter-reflections and constant albedo, the image irradiance equation becomes

$$E(x, y) = \mathbf{n} \cdot \mathbf{s}. \quad (3)$$

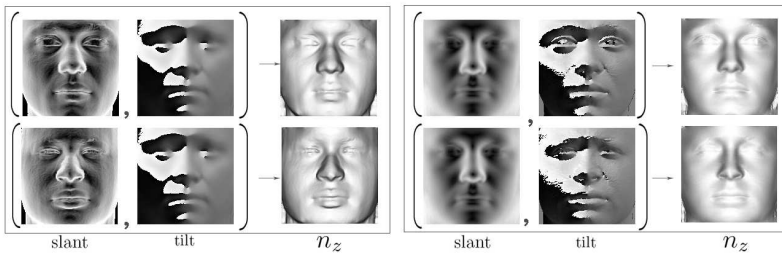
the image irradiance equation demands that the recovered surface normals lie on the reflectance cone whose axis is the light source direction and whose opening angle is the inverse cosine of the normalized image brightness. This is why SFS is an under-constrained problem: the two degrees of freedom for surface orientation (slant and tilt) must be recovered from a single measured brightness value. Surfaces rendered under Lambert's law have a matte aspect. Examples can be seen in the third column of Figure 2. The  $n_z$  component of the surface normal is indeed the Lambertian illumination of the surface, with the light source direction parallel to the viewers direction (i.e.  $\mathbf{s} = (0, 0, 1)$ ).

In contrast to the human visual system [12], it seems that computer vision systems encounter more difficulty in estimating the tilt of a surface from a single image than its slant. This is not surprising since the only measure that can be directly recovered from the (Lambertian) image brightness is the zenith angle. This fact is exploited in the next section, where we explain how the slant can be used to approximate facial surface using only a reference model and a brightness image.

## 4 Using the Zenith Angle of the Surface Normal to Approximate Facial Shape

We profit on the fact that the inverse cosine of the produced irradiance equals the zenith angle of the surface normal. This means one can calculate the zenith angle in a straightforward way. The main idea in this paper is to pair the slant calculated from a brightness image with the tilt obtained from a reference model. This will transfer the facial features of the input image onto the reference model.

In our experiments, the input brightness data was taken from the Lambertian reillumination of the subjects in the database. The face database used for building the models was provided by the Max-Planck Institute for Biological Cybernetics in Tuebingen, Germany [13]. The reference model was the average face over all the subjects in the database.



**Fig. 3.** The rows represent two different subjects (same used in Figure 2). The leftmost panel shows the case when  $\theta_{im}$  and  $\phi_{ref}$  are combined. The rightmost panel shows the case of combining  $\theta_{ref}$  and  $\phi_{im}$ .

To illustrate the combination of slant and tilt, let us call  $(\theta_{ref}, \phi_{ref})$  to the zenith and azimuth angles of the surface normal of a reference model. Similarly, let us call  $(\theta_{im}, \phi_{im})$  to the azimuth and zenith angles of an input brightness image. Unlike the zenith angle, the azimuth angle cannot be calculated directly from the image brightness, however, we assume we have accurate tilt values in  $\phi_{im}$ . In Figure 3 we show two examples. The Figure contains two panels, each of which consists of columns showing slant, tilt, and a frontal reillumination. This reillumination represents the integrated surface from the normals obtained after combining slant and tilt values<sup>1</sup>. The rows represent two different subjects (same as used in Figure 2). The leftmost panel shows the case when  $\theta_{im}$  and  $\phi_{ref}$  are combined. The rightmost panel shows the case of combining  $\theta_{ref}$  and  $\phi_{im}$ . The important feature to note here is that the main responsible of the facial appearance is the slant. In both cases, the tilt seems to provide the general shape of the face, but the slant dictates the perceptible changes in surface inclination (i.e. regions around the nose, lips, eyes).

<sup>1</sup> For surface integration from surface normals, we used the global integration method of Frankot and Chellappa [14].

## 5 Experiments

In this section, we present an experimental evaluation of the method. First, we use Lambertian data to test accuracy in surface shape recovery. Then we use real world images to generate novel reillumination and explore the usability of the recovered surfaces for face recognition applications.

### 5.1 Lambertian Examples

We performed experiments using the brightness images of each surface in the database (i.e the  $n_z$  component). The slant  $\theta_{im}$  was obtained directly from each subject and then combined with the reference tilt  $\phi_{ref}$ . A new set of surface normals was derived from the pair  $(\theta_{im}, \phi_{ref})$ . These surface normals were integrated to generate a surface. Therefore, 100 facial surfaces were approximated from each brightness image in the database.

Profile comparisons of two examples are shown in Figure 4. The ground truth (solid line) is plotted against the recovered surface (dashed line). Note how the difference in height surface is negligible, while the profile contour reveals agreement in shape.

In order to test the accuracy of facial reconstruction, we have explored the distribution of the recovered surfaces. We have used Multi Dimensional Scaling (MDS) [15] to embed the faces in a low dimensional pattern space. We built a 100 eigenfaces model based on the ground truth database and determined the dissimilarity measure in the following manner:

1. Calculate the matrix of vector coefficients for each in-training element in the database (the columns of this matrix are parameter vectors representing the in-training sample faces).
2. Calculate the linear correlation coefficient between the columns of the parameter matrix. A correlation of 1 indicates a dissimilarity of 0, a correlation of -1 indicates a dissimilarity of 1. We used only the first 40 coefficients for each vector, and these account for at least 90% of the total variance of the model. We repeated this procedure using the recovered height surfaces, building another height model from which the dissimilarity matrix was calculated in the same way as explained above.

In Figure 5 the results of performing MDS are shown as gray and white circles for the ground truth and the recovered surfaces, respectively. The distribution of dissimilarities is very similar for both set of data. This suggests that the height surfaces recovered using the method reflect the same shape distribution as the ground-truth parameters. In other words, the output of the method may be suitable for the purposes of recognition.

### 5.2 Non-lambertian Examples

Although our main interest in this paper has been the use of the the zenith angle of the surface normal for reconstructing facial shape, we have also

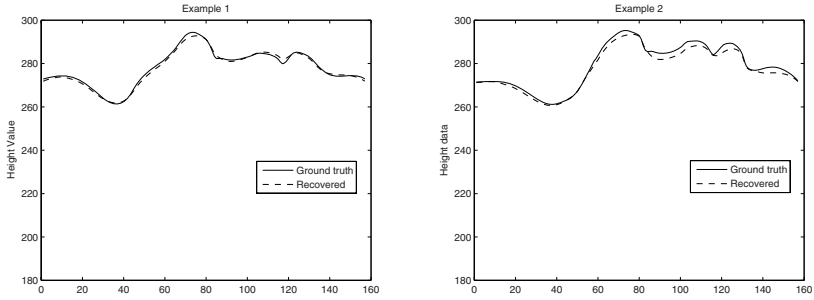


Fig. 4. The ground truth (solid line) is plotted against the recovered surface (dashed line)

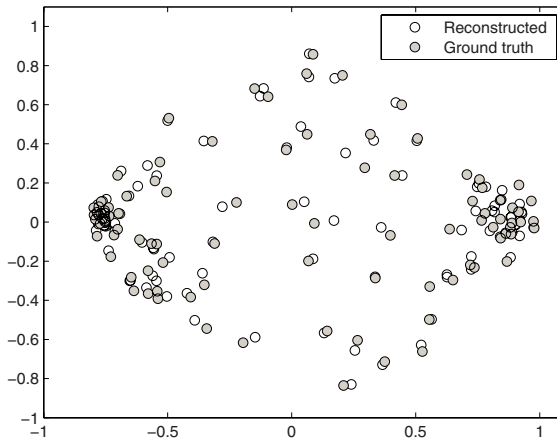


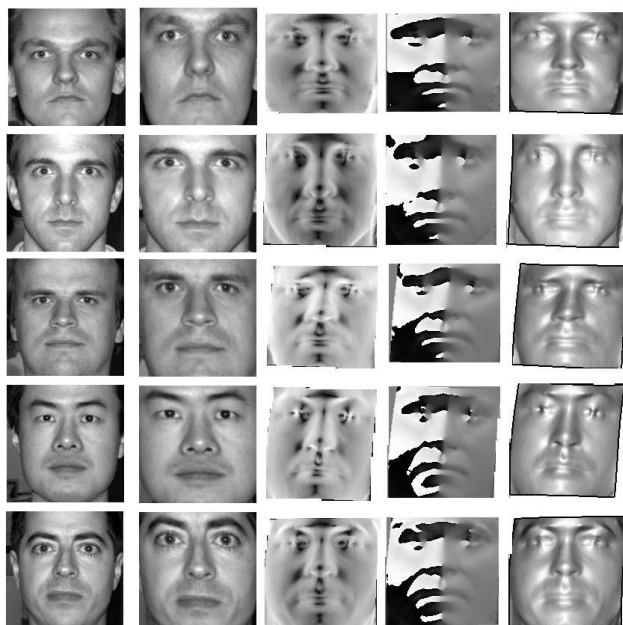
Fig. 5. Results of performing MDS are shown as gray and white circles for the ground truth and the recovered surfaces respectively

performed some relatively limited experiments aimed at exploring their potential for recognition.

We also present experiments with a number of real world face images. These images are drawn from the Yale B database [16]. In the images, the faces are in the frontal pose and were illuminated by a point light source situated approximately in the viewer direction. The input images were aligned to the reference model using a simple warping operation with 4 landmarks (the centers of the eyes, tip of the nose and center of the mouth). The recovered surfaces were un-warped to better approximate the input image.

The surface recovery results are shown in Figure 6. From left to right we show the input image, the aligned input, slant, tilt and frontal illumination corresponding to the recovered surface. Note the errors due to the non-Lambertian nature of the input images. This evidences itself as instabilities around the boundaries of





**Fig. 6.** Surface recovery results for non-Lambertian data. From left to right we show the input image, the aligned input, slant, tilt and frontal illumination corresponding to the recovered surface.

the face and in the proximity of the mouth and nose. Although the method struggled to recover the shape of the eye sockets, the overall structure of the face is well reconstructed. Moreover, the eyebrow location, nose length and width of the face clearly match those of the input images. Another important feature to note from the figure is that the recovered slant (third column) shows more correspondence with the input image than the recovered tilt (fourth column). This is because the facial features of the input images were better conserved in the slant, while the reference tilt just served as the mold where the input image was fitted.

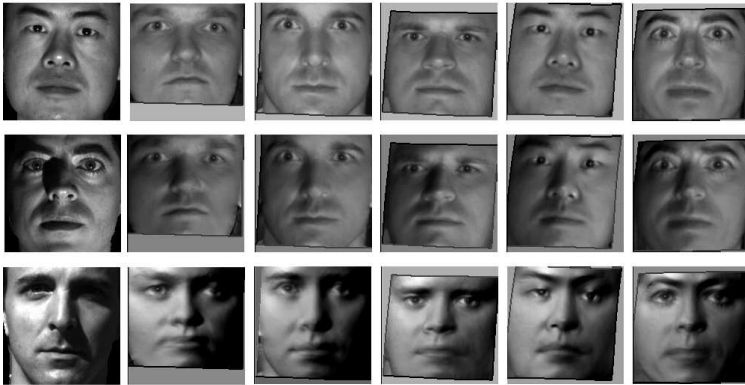
For the recognition experiments, we used the frontal view subset of the Yale B database. In the database, the subjects are illuminated from many different light sources and photographs are taken to capture the appearance of the images over a wide range of illuminations. We rendered new reilluminations and performed recognition tests following the next steps:

1. Estimation of lighting coefficients. For each subject, we used the recovered surface and a photograph. With these two estimations at hand we approximated the lighting coefficients of a spherical harmonic illumination model [17].
2. Rendering appearance. A novel reillumination is then generated using the lighting coefficients, an albedo map and the surface normals of the recovered surface.

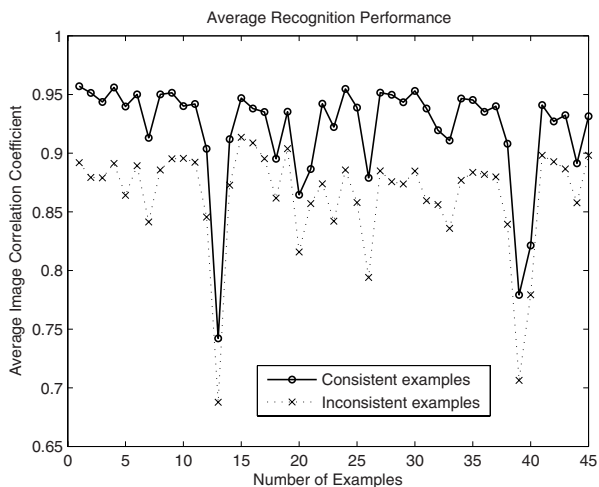
3. Comparing rendered and original image. We computed the correlation coefficient between the rendered image and the original input image. We located the element that maximized the correlation. The identity of this element was used to classify the input image. In this manner we aimed to recognize views of the subject re-illuminated from different light source directions.

In Figure 7 we show results on the rendering technique. The first column shows a sample image from which the illumination coefficients were recovered. These coefficients were used to render the recovered surfaces under similar illumination conditions. This rendering is illustrated in the remaining columns. The rows of the figure show three different illumination cases. Note that the model struggles to generate realistic views when the light source direction departs from the viewer direction. This is due to the simplicity of the rendering technique (cast shadows are not modeled) as well as the rough approximation of the surface (the bas-relief ambiguity was not considered).

Despite the simplicity of the rendering technique and the surface recovery method, the recognition tests suggest that this simple approximation can lead to favorable results. This is illustrated in Figure 8, where a plot of the subject number against the correlation coefficient achieved is shown. We computed the average behaviour for the five subjects of the database. Consistent examples (i.e. the ones supposed to get the highest correlation coefficient) are represented with a solid line. Inconsistent examples are represented with a dotted line. The figure shows how the average behaviour of the recognition test tends to assign the highest correlation coefficient to consistent examples, therefore achieving a correct classification in all the cases.



**Fig. 7.** The first column shows a sample image from which the illumination coefficients were recovered. These coefficients were used to render the recovered surfaces under similar illumination conditions. This is shown in the remaining columns. The rows show three different illumination cases.



**Fig. 8.** Consistent examples are represented with a solid line. Inconsistent examples are represented with a dotted line.

## 6 Conclusions

We have presented a method for recovering facial shape using an image of a face and a reference model. The image has to be frontally illuminated and aligned to the reference model. The zenith angle of the surface normal is recovered directly from the intensities of the image. The azimuth angle of the reference model is then combined with the calculated zenith angle in order to get a new field of surface normals. After integration of the needle map, the recovered surface has the effect of mapped facial features over the reference model. Experiments demonstrate that for the Lambertian case, surface recovery is achieved with high accuracy. For non-Lambertian cases, experiments suggest potential for face recognition applications. Future work includes using the method with more realistic rendering techniques as well as considering the bas-relief ambiguity.

## References

1. Horn, B., Brooks, M.: *Shape from Shading*. MIT Press, Cambridge (1989)
2. Jognston, A., Hill, H., Carman, N.: Recognising faces: Effects of lightning direction, inversion and brightness reversal. *Perception* 21, 365–375 (1992)
3. Atick, J., Griffin, P., Redlich, N.: Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. *Neural Computation* 8, 1321–1340 (1996)
4. Blanz, V., Vetter, T.: Face recognition based on fitting a 3d morphable model. *IEEE T. PAMI* 25(9), 1063–1074 (2003)
5. Smith, W., Hancock, E.R.: Recovering facial shape and albedo using a statistical model of surface normal direction. In: *Proc. IEEE ICCV 2005*, pp. 588–595 (2005)

6. Castelán, M., Hancock, E.R.: Using cartesian models of faces with a data-driven and integrable fitting framework. In: Campilho, A., Kamel, M. (eds.) ICIAR 2006. LNCS, vol. 4142, pp. 134–145. Springer, Heidelberg (2006)
7. Worthington, P.L., Hancock, E.R.: New constraints on data-closeness and needle map consistency for shape-from-shading. *IEEE T. PAMI* 21(12), 1250–1267 (1999)
8. Kemelmacher, I., Basri, R.: Molding face shapes by example. In: Proc. European Conference in Computer Vision (2006)
9. Marr, D.: *Vision: A Computational Investigation into the Human Representation and Processing of the Visual information*. Freeman (1982)
10. Moses, Y., Adini, Y., Ullman, S.: Face recognition: the problem of compensating for changes in illumination direction. In: Proc. European Conference on Computer Vision, pp. 286–296 (1994)
11. Horn, B.: Understanding image intensities. *Artificial Intelligence* 8, 201–231 (1997)
12. Erens, R., Kappers, A., Koenderink, J.: Perception of local shape from shading. *Perception and Psychophysics* 54(2), 145–156 (1993)
13. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH 1999, pp. 187–194 (1999)
14. Frankot, R., Chellappa, R.: A method for enforcing integrability in shape from shading algorithms. *IEEE T.PAMI* 10, 438–451 (1988)
15. Young, F.W., H.R.M.: *Theory and Applications of Multidimensional Scaling*. Eribaum Associates, Hillsdale (1994)
16. Georghiadis, A., Belhumeur, D., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. In: *IEEE T. PAMI*, pp. 634–660 (2001)
17. Basri, R., Jacobs, D.W.: Lambertian reflectance and linear subspaces. *IEEE T. PAMI* 25(2), 218–233 (2003)

# Feature Extraction and Face Verification Using Gabor and Gaussian Mixture Models

Jesus Olivares-Mercado, Gabriel Sanchez-Perez, Mariko Nakano-Miyatake,  
and Hector Perez-Meana

ESIME Culhuacan, Instituto Politécnico Nacional, Av. Santa Ana No. 1000, Col. San Francisco  
Culhuacan, 04430 Mexico D.F. Mexico  
mariko@calmecac.esimecu.ipn.mx

**Abstract.** This paper proposes a faces verification in which the feature extraction is carried out using the discrete Gabor function (DGF), while the Gaussian Mixture Model (GMM) is used in the face verification stage. Evaluation results using standard data bases with different parameters, such as the number of mixtures and the number of face used for training show that proposed system provides better results that other proposed systems with a correct verification rate larger than 95%. Although, as happens in must face recognition systems, the verification rate decreases when the target faces present some rotation degrees.

**Keywords:** Gabor functions, Gaussian Mixture Model, Face verification.

## 1 Introduction

The development of security systems based on biometric features has been a topic of active research during the last three decades, because the verification of the people identity to access control and to enforce security in restricted areas, etc. are fundamental aspects in these days. The terrorist attacks happened during the last decade have demonstrated that it is indispensable to have reliable security systems in offices, banks, companies, trades, etc.; increasing in such way the necessity to develop more reliable methods to verify the people identity. The identity verification systems using biometric methods appear to be good alternatives for the development of such security systems.

The biometrics systems consist of a group of automated methods for recognition or verification of people identity using physical characteristics or personal behavior of the person under analysis [1]. This technology is based on the premise that each person is unique and possesses distinctive features that can be used to identify him. Following these ideas several biometric based security systems have been developed using fingerprints, iris, voice, hand and face features. Among them, the face verification systems appear to be a desirable alternative because it is non-invasive and its computational complexity is relatively low.

The face verification has been a topic of active research during the last three decades; because it is perhaps, the biometric method easier of understanding because for us the face is the most direct way to identify the people. In addition the data

acquisition of this method consists in taking a picture, doing it one of the biometric methods with larger acceptance among the users.

The recognition is a very complex activity of the human brain. For example, we can recognize hundred of faces learned throughout our life and to identify familiar faces at the first sight, even after several years of separation, with relative easy. However for a computer it is not a simple task. For instance, recently proposed face recognition systems, achieve a recognition rate of about 91% when the face image is not rotated or the rotation is relatively low. However although, this recognition rate is good enough for several practical applications, it may be not large enough for applications where the security should be extreme; such that we cannot tolerate a high erroneous recognition rate. This paper proposes a face recognition algorithm that is able of achieving an erroneous verification rate below 9%.

Several methods have been proposed for face recognition [2], [3], such as the methods of the statistical correlation of the face geometry [4]; the face form which uses the distances among the position of the eyes, mouth, nose, etc. as well as those using the neuronal networks technology that trait to imitate the operation of the human brain. Many of these systems can recognize a person even when they present some physical changes, such as the growth of the beard or mustache, changes in the color or hair style, the use of glasses, etc. Although in general they are sensitive to rotations of the face images.

Before beginning to analyze the procedures used for face recognition, it is necessary to point out the verification concept. In face verification, the person informs to the system about his/her identity, presenting an identification card or writing a special password, etc. The system captures the person's features (for example the persons' face in this case), and the proceeds to determine if the people is whom his/her claims to be.

## 2 Proposed System

This section provides a detailed description of the proposed face verification algorithm which consists of three stages. Firstly a feature extraction of the face is

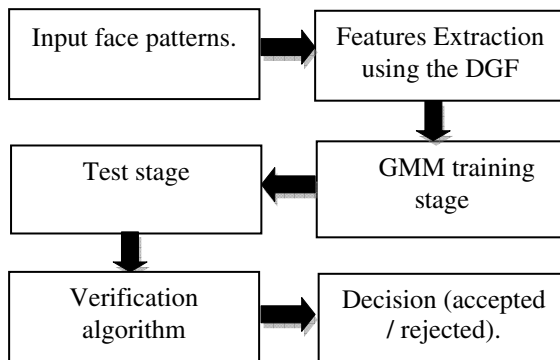
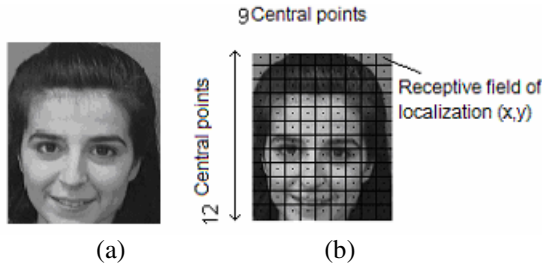


Fig. 1. Proposed face verification algorithm



**Fig. 2.** a) Original face. b) Face image divided in 108 receptive fields and 108 central points (x, y).

carried out using the Gabor discrete transform (DGF). Next using these feature vectors, a model for each face is obtained using a Gaussian Mixture Model (GMM). Finally during the verification process, the GMM output is used to take the final decision. Figure 1 shows the block diagram of proposed algorithm.

### 2.1 Feature Extraction Stage

The feature extraction plays a very important role in the any pattern recognition system. To this end, the proposed algorithm uses the DGT which has some relation with the human visual system (HVS). The two dimensional discrete Gabor functions (2D-DGF) depend on four parameters, two that express their localization in the space (x, y) and other two that express the spatial frequency,  $f_m$ , and the orientation  $\phi_n$ , with  $m=1,2,..N_f$  and  $n=1,2,..,N_\phi$  [5]. Thus to estimate the features vector, firstly the captured image (NxM) is divided in  $M_x M_y$  receptive fields each one of size  $(2N_x+1) \times (2N_y+1)$  (Fig. 2), where  $N_x=(N-M_x)/2M_x$ ,  $N_y=(M-M_y)/2M_y$ . This fact allows that the features vector size be independent of the captured image size. Next, the central point of each receptive field whose coordinates are given by  $(c_i, d_k)$ , where  $i=1,2,..,N_x$ ;  $k=1,2,3,..,N_y$ , are estimated. Subsequently it is estimated the first point of the cross-correlation between each receptive field and the  $N_f N_\phi$  Gabor functions which are given by

$$h_{m,\phi}(x, y) = g(x', y') \exp(j2\pi f_m(x'+y')) \tag{1}$$

where

$$(x', y') = ((x \cos \phi_n + y \sin \phi_n), (-x \sin \phi_n + y \cos \phi_n)) \tag{2}$$

The DGF, which are complex valued functions, can be represented as

$$h_{m,\phi}(x, y) = h_{m,\phi}^c(x, y) - j h_{m,\phi}^s(x, y) \tag{3}$$

where

$$h_{m,\phi}^c(x, y) = g(x', y') \cos(2\pi f_m(x'+y')) \tag{4}$$

$$h_{m,\phi}^s(x, y) = g(x', y') \sin(2\pi f_m(x'+y')) \tag{5}$$

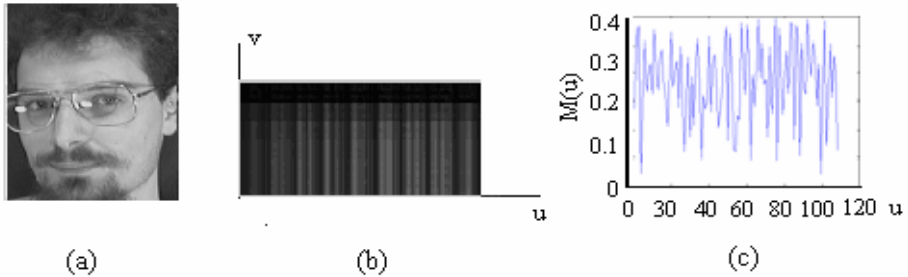
Equations (4) and (5) denote the real and imaginary DGF components. Next using eqs.(3)-(5) we can estimate the first component of the cross correlation function of each DGF with each receptive field as follows

$$\psi(u,v) = \sum_{x=-N_x}^{N_x} \sum_{y=-N_y}^{N_y} I(x-c_i, y-d_k) \left( h_{m,\phi}^c(x-c_i, y-d_k) - h_{m,\phi}^s(x-c_i, y-d_k) \right) \quad (6)$$

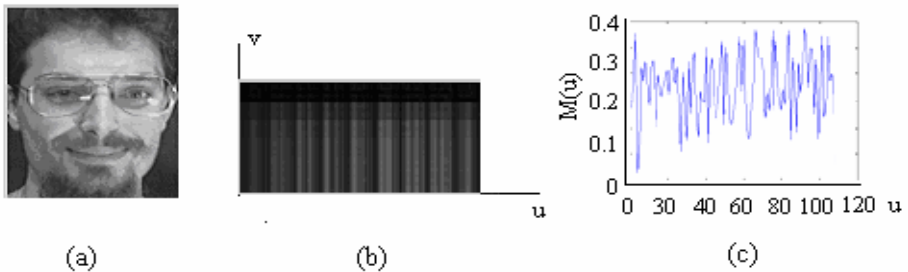
where  $u=M_y(i-1)+k$  and  $v=N_f(m-1)+n$ . Next, to avoid complex valued data in the features vector we can use the fact that the magnitude of  $\psi(u,v)$  presents a great similarity with the behavior of the complex cells of the human visual system. Thus the magnitude of  $\psi(u,v)$  could be used. However, as shown in eq.(6) the number of elements in the features vector is still so large even for small values of  $M_v$ ,  $M_y$ ,  $N_\phi$  y  $N_f$ . Thus to reduce the number of elements in the features vector, we can use the first point of the total cross correlation between each receptive field and the set of DGF, which can be obtained by taking the average of  $\psi(u,v)$  with respect to  $v$ . Thus the features vector of proposed algorithm,  $M(u)$ , becomes

$$M(u) = \frac{1}{N_v} \sum_{v=1}^{N_v} |\psi(u,v)| \quad (7)$$

Where  $N_v=N_fN_\phi$ . Figure 3 illustrate this procedure.



**Fig. 3.** a) Original image. b) Matrix containing the first point of cross correlation between DGF and receptive fields. c) Features vector of proposed algorithm.



**Fig. 4.** a) Original image. b) Original image. b) Matrix containing the first point of cross correlation between DGF and receptive fields. c) Features vector of proposed algorithm.



### 2.2 Face Verification Stage

To perform the face verification task a GMM will be used because, the GMM, which consists of a sum of M weighted Gaussian density functions is able to approximate any probability distribution if the number of Gaussian components is large enough. Consider the GMM shown in Fig. 5 which is described by the following equation [6]:

$$p(\vec{x} | \lambda) = \sum_{i=1}^M p_i b_i(\vec{x}) \tag{8}$$

where  $\vec{x}$  is a N-dimensional vector,  $b_i(\vec{x}), i=1,2,\dots,M$ , are the density components and  $p_i, i=1,2,\dots,M$ , are the mixture weights. Each density component is a D-dimensional Gaussian function given as:

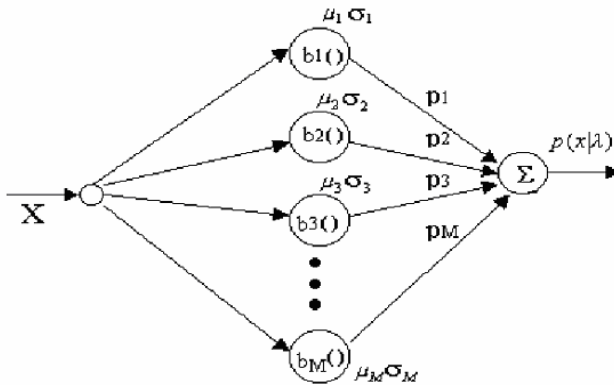


Fig. 5. Gaussian Mixture Model

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} |\sigma_i|^{1/2}} \exp \left\{ -\frac{1}{2} (\vec{x} - \vec{\mu}_i)' \sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \right\} \tag{9}$$

where ( )' denotes the transpose vector,  $\mu_i$  is the mean vector and  $\sigma_i$  the covariance matrix which is assumed to be diagonal; and  $p_i$  denotes the mixture weights which satisfy that:

$$\sum_{i=1}^M p_i = 1 \tag{10}$$

The distribution model is determined by the mean vector, the covariance matrix and the distribution weights such that that the model of the face under analysis is given by

$$\lambda = \{p_i, \mu_i, \sigma_i\} \quad i = 1, 2, \dots, M \tag{11}$$

The optimal parameter estimation is a non-linear problem, such that we need an iterative algorithm to estimate the optimal parameters of the face verification algorithm. Thus we can use the ML algorithm (Maximum Likelihood) to search for the parameters of the system providing the best approach to the model of the face under analysis. That is the target is to find the parameters of  $\lambda$  that maximize the a posteriori probability distribution. For a sequence of vectors  $T$  of training  $X=\{x_1, \dots, x_T\}$ , GMM likelihood can be written as:

$$p(X/\lambda) = \prod_{t=1}^T p(X_t/\lambda) \tag{12}$$

Unfortunately the equation (12) is non-linear in relation with the parameters  $\lambda$ . However, these can be estimated in an iterative way using the EM algorithm (Expectation-Maximization), in which starting from an initial set of parameters  $\lambda(r-1)$  a new model is estimated  $\lambda(r)$ , where  $r$  denotes the  $r$ -th iteration, such that:

$$p(X/\lambda(r)) \geq p(X/\lambda(r-1)) \tag{13}$$

To accomplish this task, each  $T$  elements the GMM parameters of are updated as follows

**Mixture weights:**

$$p_i = \frac{1}{T} \sum_{t=1}^T p(i/X_{t+k}, \lambda) \tag{14}$$

**Mean:**

$$\mu_i = \frac{\sum_{t=1}^T p(i/X_{t+k}, \lambda) X_{t+k}}{\sum_{t=1}^T p(i/X_{t+k}, \lambda)} \tag{15}$$

**Covariance:**

$$\sigma_i = \frac{\sum_{t=1}^T p(i/X_{t+k}, \lambda) (X_{t+k} - \mu_i)^2}{\sum_{t=1}^T p(i/X_{t+k}, \lambda)} \tag{16}$$

Finally the probability to posteriori it is obtained for:

$$p(i/X_{t+k}, \lambda) = \frac{p_i b_i(X_{t+k})}{\sum_{j=1}^M p_j b_j(X_{t+k})} \tag{17}$$

During the testing phase we need to estimate the probability that the face under analysis corresponds to a given model, that is  $P_r(\lambda|X)$ . To this end consider the Bayes theorem which is given by

$$\hat{R} = \Pr(\lambda / X) = \frac{p(X / \lambda)\Pr(\lambda)}{p(X)} \tag{18}$$

where  $P(X/\lambda)$  is the given by eq. (12),  $P_r(\lambda)$  is the probability distribution of  $\lambda$  model, and  $P(X)$  is the probability distribution the face under analysis. Assuming that all faces are equally probable then  $Pr(\lambda)=1/R$ . Next taking in account that  $P(X)$  is same for all the face models, and substituting eq. (12) into eq. (18) it follows that

$$\hat{R} = p(X / \lambda) = \prod_{t=1}^T p(X_t / \lambda) \tag{19}$$

Finally taking the logarithm of eq. (19) we get

$$\hat{R} = \sum_{t=1}^T \log_{10}(p(X_t / \lambda)) \tag{20}$$

where  $p(X/\lambda)$ , which is given by (12), is the output of GMM system shown in the Fig 5.

### 3 Evaluation Results

The evaluation of proposed system was carried out by computer simulations using the database created by Olivetti Research Laboratory in Cambridge, UK (ORL), which consists of images of 30 people with 10 images of each one which differs on illumination, face rotation, different inclination, different hairstyle, wardrobe changes, etc. The images size is 128 x 128 pixels.

To do the proposed method robust against changes of sizes and translation; the algorithm firstly assumes that the gray level of picture background is constant. Next the algorithm estimates the position and size of the image by analyzing the gray label variation on the image. Once the image size and position have been estimated, the image is divided in 12x9 receptive fields, as shown in Fig. 2, whose central point will be always located in the space position (x, y), where x=0 and y=0. After the image was divided in 108 receptive, the features vector is estimated using eqs. (1)-(6) with 9 phases 0,  $\pi/9$ ,  $2\pi/9$ ,  $\pi/3$ ,  $4\pi/9$ ,  $5\pi/9$ ,  $2\pi/3$ ,  $7\pi/9$  and  $8\pi/9$ ; and six normalized frequencies  $\pi/2$ ,  $\pi/4$ ,  $\pi/8$ ,  $\pi/16$ ,  $\pi/32$ ,  $\pi/64$ . This produces a matrix with 5832 elements that are subsequently reduced to 108 using eq. (7), which are different and unique for each person and they are similar for the same person like it is shown in the figure 3 and 4.

Next the feature vector is it is applied to a GMM to obtain the model of the face under analysis. This is achieved substituting the feature vectors obtained in the Gaussian mixture model (GMM) to estimate the weights, the mean and variance as described by eq. (9)-(17). To training the GMM we assume that the 108 elements

estimated in the feature extraction stage are divided in features vector of L elements as follows

$$S = \{X_0, X_1, X_2, \dots, X_T, X_{T+1}, X_{T+2}, X_{T+3}, \dots, X_{L-1}\} \tag{21}$$

Subsequently, using these vectors a group of vectors in L segments with T features vectors,  $X_t$ , each one, is formed in the following way:

$$S_0 = \{X_0, X_1, X_2, X_t, X_{t+1}, \dots, X_T\} \tag{22}$$

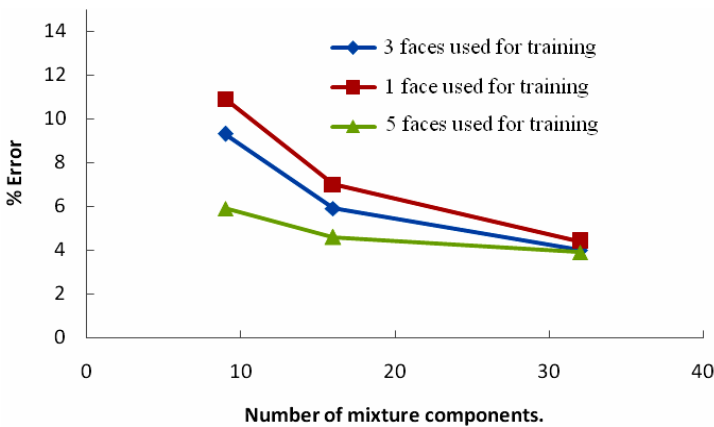
\*

$$S_k = \{X_k, X_{k+1}, X_{k+2}, \dots, X_t, X_{t+1}, \dots, X_{T+k}\} \tag{23}$$

In this work we take T=12 in order that each 12 feature vectors the GMM parameters be updated.

**Table 1.** False rejection error

# of faces of training	# Gaussian Mixtures	Low threshold %	Half threshold %	high threshold %
1 face	9	12	8	5.1
	16	18.5	12.9	8.7
	32	12.4	9.2	6.8
3 faces	9	12	7.7	4.7
	16	13.25	9.1	5.9
	32	8.66	6.48	4.6
5 faces	9	9.3	5.9	4
	16	10.9	7	4.4
	32	5.9	4.6	3.9

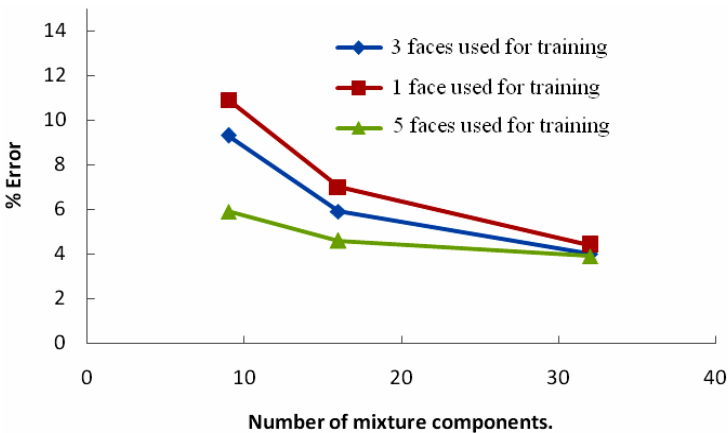


**Fig. 6.** Rejection error with different number of mixtures and 1, 3 and 5 training faces of each person

In the verification stage a threshold is used which depends of the face under analysis. To improve de verification performance this threshold will be divided in three categories: low, half and high thresholds. Two additional variants were also introduced for evaluation: one is the numbers of faces used for the training and the second one is the numbers of Gaussians mixtures to be used. Here the numbers of faces used are 1, 3, and 5; each one with 9, 16 and 32 Gaussian mixtures. Table I shows the results obtained using the database of 300 faces. These results correspond to the average error obtained during the test phase. Figure 6 shows that better results can be obtained when the number of mixtures and images used for training increase. Here is evident that in this situation the error decreases considerably using any of the three proposed thresholds.

**Table 2.** False acceptance error

# of faces of training	# Gaussian Mixtures	Low threshold %	Half threshold %	high threshold %
1 face	9	11.57	6.71	3.14
	16	18.5	12.35	7.7
	32	11.8	8.34	5.62
3 face	9	11.4	6.5	3.13
	16	12.5	8	4.5
	32	7.42	5	2.9
5 face	9	8.7	4.5	2.2
	16	10.2	5.8	2.9
	32	4.4	2.8	2



**Fig. 7.** False acceptance error with different number of mixtures and 1, 3 and 5 training faces of each person

Simulation results show that proposed algorithm performs fairly well in comparison with other previously proposed methods [1], [2], [6], even with faces that present an appreciable rotation, as happens in the ORL database.

## 4 Conclusions

This paper proposed a face verification algorithm in which the Gabor functions are used for feature extraction and the GMM to perform the verification task. Evaluation results shows that the system achieves a false rejection error between 18.5% in the worst case cases to 3.9% cases at best, when a different number of images for training as well as different number mixtures in the GMM are used; while the false acceptance error obtained is between 18.5 and 2; depending on the number of mixtures and number of faces used for training. These results can be considered to be fairly good if we consider that the database used is composed by 300 different faces. This quantity of face is very similar to any database in a real application. Evaluation results show that the system performance increase when more faces are used for training the GMM and it has a larger number of mixtures. This is valid for false acceptance error as well as for false rejection error.

## Acknowledgements

We thanks the National Science and Technology Council and to the National Polytechnic Institute for the financial support during the realization of this research.

## References

- [1] Reid, P.: *BIOMETRICS for Networks Security*, pp. 3–7. Prentice Hall, New Jersey (2004)
- [2] Chellappa, R., Wilson, C., Sirohey, S.: Human and Machine Recognition of Faces: A Survey. *Proc. IEEE* 83(5), 705–740 (1995)
- [3] Shashua, A.: *Geometry and Photometry in 3D Visual Recognition*. PhD thesis, Massachusetts Institute of Technology (1992)
- [4] Baron, R.J.: Mechanisms of human facial recognition. *International Journal of Man-Machine Studies*, 137–178 (1981)
- [5] Dunn, D., Higgins, W.E.: Optimal Gabor Filters for Texture Segmentation. In: *IEEE Trans. Image Proc* 4(7) (July 1995)
- [6] Kim, J.Y., Ko, D.Y., Na, S.Y.: Implementation and Enhancement of GMM Face Recognition Systems Using Flatness Measure. *IEEE Robot and Human Interactive Communication* (September 2004)
- [7] Reynolds, D.A., Rose, R.C.: Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. *IEEE Trans. Speech and audio Proc* 3(1) (January 1995)

# Lips Shape Extraction Via Active Shape Model and Local Binary Pattern

Luis E. Morán L. and Raúl Pinto-Elías

Centro Nacional de Investigación y Desarrollo Tecnológico  
Interior internado Palmira s/n, Cuernavaca, Morelos, 62490 México  
{lem172, rpinto}@cenidet.edu.mx

**Abstract.** In this work we assume a frontal view of a face for the lips shape extraction, then the first step is locate a face inside a digital image, for this task we use techniques based in color to extract only the pixels with skin tone, a templates based in integral projections are applied to verify and locate the face, using integral projections, we locate and define a region of interest for lips. Previously a statistical model of lips (ASM) was created in the same way, local appearance patterns of landmarks are modeled using Local Binary Patterns (LBP), in this model we try to capture a variation from a closed lips to an opened lips. For the search task Local Binary Pattern Histogram (LBPH) are used.

## 1 Introduction

In the context of the speech recognition, there are troubles when the environment is very noisy in some cases auditive information is totally null, has been shown that the visual information obtained of an conversation is useful for improving the accuracy of speech recognition in both humans and machines, this characteristic is demonstrated by the "McGurk effect" which can be explained as the complementary use of acoustic and visual information for speech perception[15]. The visual information obtained of an conversation is given by the lips, teeth and tongue, complementary visual information can be extracted of the eyes and eyebrows.

In this work the lips shape is obtained from digital images. The image is segmented in areas with skin tone pixels, we assume that a face is the biggest area with skin color. An area is declared a face according to the calculus of the integral projections from this area and the alignment with a few templates. To align the integral projections with templates Dynamic time warping [8] is used. In the above process three templates are used, the first template is for the vertical projection of the whole area, the second template is for the horizontal projection of the eyes area and finally the third template is used in the horizontal projection of the mouth area. With the information about integral projection, the Active Shape Model is initialized. The Local Binary Pattern histograms are used in the search task of the ASM.

The section 2 describes the method for face localization and lips region extraction, using integral projections and dynamic time warping. The extraction of the lips shape are presented in the section 3. The results are presented in section 4 and finally the conclusions are presented in the section 5.

## 2 Face Localization and Lips Area Extraction

The first step is to segment the image in regions with skin tone. Skin detection consists in eliminate non-skin pixels in an image, a decision rule is used for this task. We use a technique that defines explicitly the boundaries of skin tone cluster in RGB color space [5].

Three templates based on integral projections from an average face, are used to validate if a region in the image is a face, the validation is carried out aligning the projections calculated to an image, with the previously created templates, we provide an algorithm using Dynamic Time Warping to align these projections.

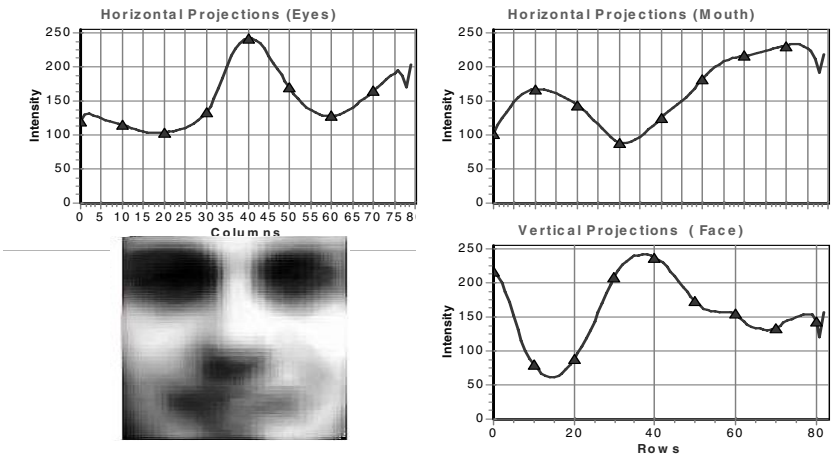
In a grayscale image, an integral projection gives a marginal distribution of the gray values along one direction, vertical or horizontal. The vertical and horizontal integral projections are given by:

$$H(x) = \frac{1}{N_x} \sum_{y=1}^N i(x, y) \tag{1}$$

$$V(y) = \frac{1}{N_y} \sum_{x=1}^N i(x, y) \tag{2}$$

### 2.1 The Templates

A set of images that contain only faces is used to generate an average face, this average face is equalized and filtered with a median filter, then the integral projections are calculated, they will be the templates. A good representation of the face is given by one vertical projection and two horizontal projections [3]. The vertical projection contains the full area of the face, one horizontal projection for the eyebrows and eyes and other for the nose and mouth.



**Fig. 1.** Average face from 66 images, Vertical and horizontal projections of average face

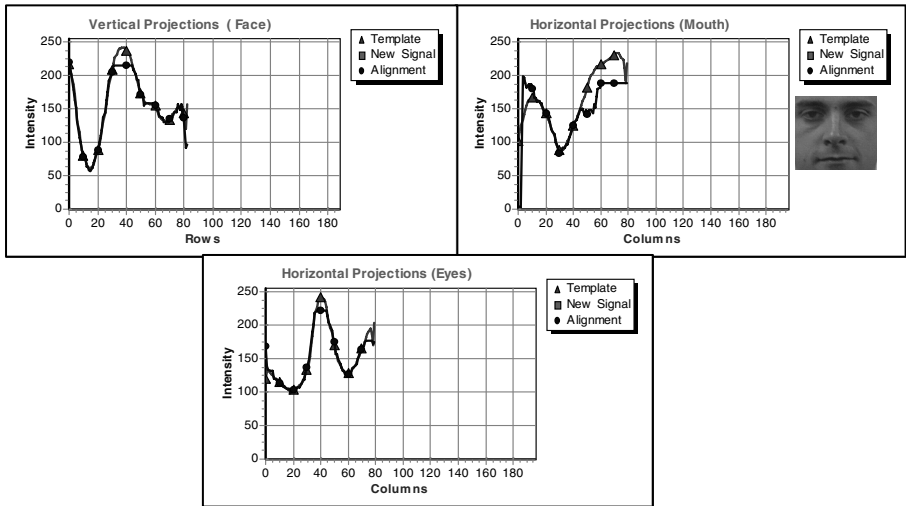


## 2.2 Alignment and Face Verification

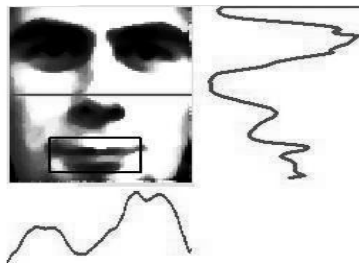
Dynamic Time Warping has been designed for aligning one curve with respect to another [8]. This technique provides a procedure to align optimally in two series of time and give the average distance associated with the optimal warping path [7]. DTW is able to handle series with unequal length, this feature is useful in pattern recognition because it is necessary to handle an object in different scales.

The result of the alignment is used to decide if the region with skin color pixels is a face. The similarity is measured, through of the pearson's coefficient correlation and euclidean distance between the warped signal and the template.

The figure 2 show a good alignment when a face is present in a region of any image, a) the skin color region containing the face, b) new projection with square shape and template with triangle shape, note that the new projection is bigger that template, c) the alignment between the two series is represented with circle shape.



**Fig. 2.** Region of the face in grayscale and its projections alignment with DTW



**Fig. 3.** Lips region extraction

### 2.3 Lips Region Extraction

The next step is extract the region where the lips are located, the face image is divided in two parts of equal dimensions the bottom part correspond at the area of the mouth in a frontal view of a face. With the information of the vertical projection and the horizontal projection of the mouth area, the limits of the lips region are established.

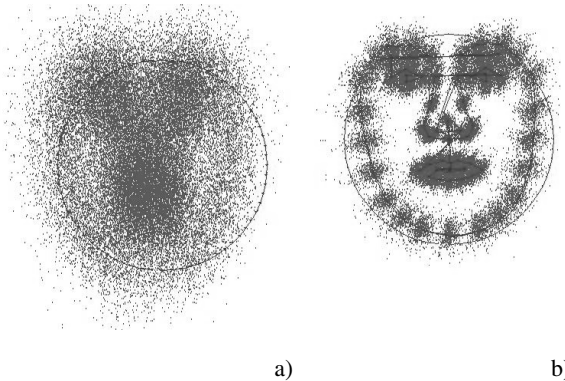
## 3 Lips Shape Extraction

By considering different approaches, we decided to employ Active Shape Model with LBP to modeling the appearance.

### 3.1 Model of the Lips

Active Shape Models are statistical models of the shapes of objects. The shape of an object is represented by a set of  $n$  points, which may be in any dimension. Commonly the points are in two or three dimensions [16].

To obtain a statistical description of the shape and variation of the object, we start with a training set, each element of the training set has a quantity of points, these points are aligned to obtain a convenient rotation, scale and translation for each mode, so that the sum of distances of each shape to the mean is minimized  $D = \sum |x - \bar{x}|^2$



**Fig. 4.** a) Unaligned points, b) Aligned points and mean shape

To manipulate in a easy way these data, it's better to reduce the dimension of them. An effective approach is to apply Principal Components Analysis (PCA)[16].

If PCA is applied to the data, then is possible to approximate any of the training set,  $x$  using :

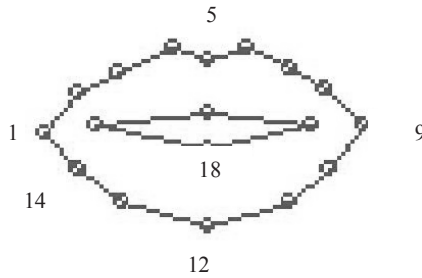
$$x \approx \bar{x} + Pb_b \quad (3)$$

Where  $\bar{x}$  is the mean shape ,  $P = (p_1|p_2|\dots|p_t)$  contains t eigenvectors of the covariance matrix and  $\mathbf{b}$  is a t dimensional vector given by :

$$\mathbf{b} = P^T (x - \bar{x}) \quad (4)$$

It should be noted that only the first few eigenvectors corresponding to the largest eigenvalues are sufficient for modeling the shape variation.

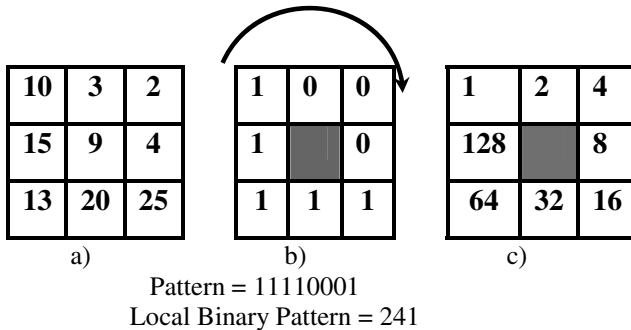
In this work 18 points are used to describe the lips shape, the points are labeled in clockwise direction as shown in the figure 5.



**Fig. 5.** Lips model representing the inner with four points and outer contour with fourteen

### 3.2 Local Binary Patterns

The local binary pattern (LBP) texture analysis operator is defined as a gray-scale invariant texture measure[16], derived from a general definition of texture in a local neighborhood. The original LBP operator was introduced by Ojala [16], the operator labels the pixels of an image by thresholding the  $3 \times 3$ -neighborhood of each pixel with the center value and considering the result as a binary number [16].



**Fig. 6.** Calculating the LBP code, a) original gray values, b) thresholded values, c) weights matrix

At a given pixel position  $(x, y)$ , the decimal value is given by:

$$LBP(x, y) = \sum_{n=0}^7 s(i_n - i_c)2^n \tag{5}$$

where  $i_c$  is the gray value of the pixel central and  $i_n$  corresponds to the gray values of the 8 surrounding pixels, the function  $s(x)$  is defined as :

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{6}$$

### 3.3 Local Appearance Modelling

To extract the information about of the gray level structures, a square centered in at landmark of ASM is used to extract information about of the gray level structures., the square has a dimension of 9 pixels by side, for each training image there is a vector  $\mathbf{X}$  to represent a planar shape of the lips.

$$\mathbf{X} = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)^T \tag{7}$$

Where  $(x_j, y_j)$  are the coordinates of the  $j^{\text{th}}$  landmark, then for each landmark we construct an LBP histogram in the area defined for a 9x9 square, finally a mean LBP histogram is computed using :

$$\bar{H}_{i,j} = \frac{1}{n} \sum_n H_{n,i,j} \tag{8}$$

Finally we have a mean LBP histogram for each landmark in the Active Shape Model, this information is used in the search task to find the lips shape in an image.

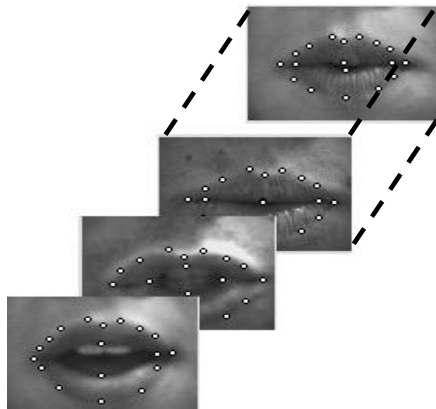
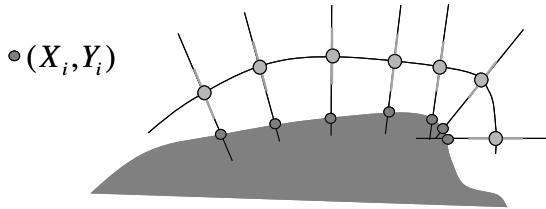


Fig. 7. Calculating the LBP histogram for each landmark

In the search task, the LBP histogram is calculating for the points along normal to boundary, then this histogram is compared with the mean histogram [17], The similarity between the histograms is measured using the chi square statistic [17].

$$\chi^2(H, \bar{H}) = \sum_i \frac{(H_i - \bar{H}_i)^2}{(H_i + \bar{H}_i)} \quad (9)$$



**Fig. 8.** Search along normal to boundary

## 4 Results

### 4.1 Data Preparation and Implementation

We have implemented this method in Builder C++ 5.0. To obtain the files of points, we use the tools of Tim Cootes, this tools are available on internet. The model of the lips was learned from Tulips1 database [6], consists of each of 12 subjects saying the first four digits of English twice, we selected 34 images with the most representative shapes of the lips. The model was tested with a set of 47 images that was captured in previous projects and 16 images extracted from internet, with the particularity that they be human faces with a frontal view.

### 4.2 Experimental Results

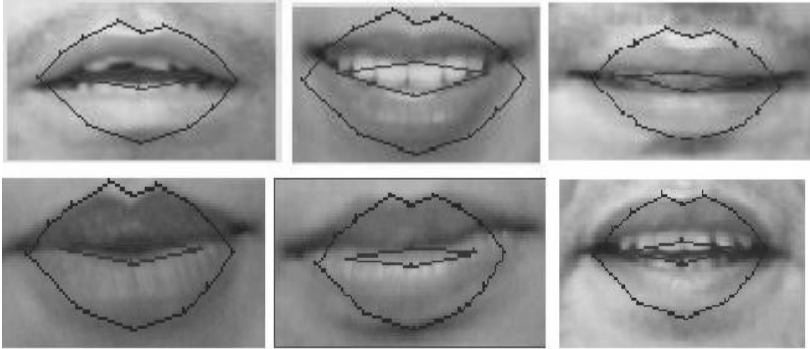
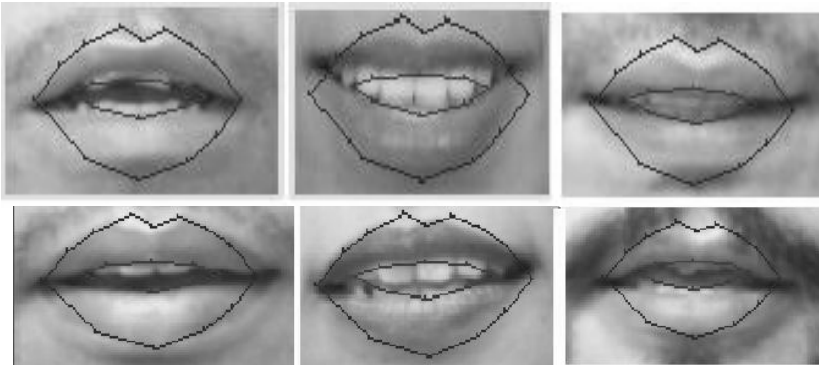
In our experiments the initialization of the shape in the image is using the information obtained from the integral projections, with this information we locate the corners and the upper and lower contour of the lips, the initial shape is placed surrounded at these points. All the failed cases are associated with poor contrast between the lips and the skin that surrounded the lips.

The main objective of this project is to recognize the shape of the lips in the image. We construct 2 models of the lips for our experiments, one model consist of 18 points for the inner and outer contour and the other model consist of 22 points for the inner and outer contour. The table 1 shows the performance of both models, the three first modes of the models are used because they represent a variability of 77.8 %.

The figures 9 shows the some results of the convergence of the model with 18 points, although some points in the second image are far of the lips boundary, the inner contour of the model is in the correct place, this result give us a good recognition of the lips shape, the same result is for the model with 22 points in the figure 10.

**Table 1.** Performance of the models using 63 images

<b>Model</b>	<b>Av. Recognition</b>
18_Points_Model	80.56 %
22_Points_Model	74.7 %

**Fig. 9.** Results with model of 18 points**Fig. 10.** Results with model of 22 points

## 5 Conclusions

In this paper we present a shape lips extraction method using LBP and ASM, the results obtained are good, but we think that it is possible to improve these results.

For modeling the variation in the shape of the lips, is not necessary a big amount of samples, just choose images of lips that represent the different articulations during a conversation.

The next work is to create transitive models based on HMM, for a group of syllables in the Mexican Spanish language, as well as to develop a system for lipreading.

## References

1. Gandhi, A.: Content-Based Image Retrieval: Plant Species Identification. Master's thesis, Oregon State University (September 2002)
2. Xu, C., Prince, J.L.: Gradient Vector Flow: A New External Force for Snakes. In: Proc. IEEE Conf. on Comp. Vis. Patt. Recog (CVPR), pp. 66–71. Comp. Soc. Press, Los Alamitos (1997)
3. Mateos, G.G., García, A.R., Lopez-de-Teruel, P.E.: Face Detection Using Integral Projection Models. In: Caelli, T.M., Amin, A., Duin, R.P.W., Kamel, M.S., de Ridder, D. (eds.) SPR 2002 and SSPR 2002. LNCS, vol. 2396, pp. 6–9. Springer, Heidelberg (2002)
4. Mateos, G.G.: Refining Face Tracking with Integral Projections. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 9–11. Springer, Heidelberg (2003)
5. Kovac, J., Peer, P., Solina, F.: Human Skin Colour Clustering for Face Detection. In: Zajc, Baldomir (eds.) EUROCON 2003 -International Conference on Computer as a Tool, Ljubljana, Slovenia (2003)
6. Movellan, J.R.: Visual Speech Recognition with Stochastic Networks. In: Tesauro, G., Touretzky, D., Leen, T. (eds.) ANIPS, vol. 7, pp. 851–858. The MIT Press, Cambridge (1995)
7. Paliwal, K.K., Agarwal, A., Sinha, S.S.: A modification over Sakoe and Chiba's dynamic time warping algorithm for isolated word recognition. *Signal Processing* 4(4), 329–333 (1982) (also in Proc. IEEE Intern. Conf. on Acoustics, Speech and Signal Processing, Paris, France, pp. 1259–1261, 1982)
8. Wang, K., Gasser, T.: Alignment of curves by dynamic time warping. *Annals of Statistics* 25(3), 1251–1276 (1997)
9. Fan, L., Sung, K.K.: Face Detection and Pose Alignment Using Color, Shape and Texture Information, vs. In: Third IEEE International Workshop on Visual Surveillance (VS'2000), p. 19 (2000)
10. Nordstrøm, M.M., Larsen, M., Sierakowski, J., Stegmann, M.B.: The IMM Face Database—An Annotated Dataset of 240 Face Images, technical report, Informatics and Mathematical Modelling, Technical Univ. of Denmark, DTU (May 2004)
11. Psychological Image Collection at Stirling (PICS), (last access in 15/12/, 2006), Available at <http://pics.psych.stir.ac.uk/>
12. Niels, R.: Dynamic Time Warping. An Intuitive Way of Handwriting Recognition. Master's thesis, Radboud University Nijmegen, Faculty of Social Sciences, Department of Artificial Intelligence / Cognitive Science, Nijmegen, The Netherlands (2004)
13. Hsu, R.-L., Abdel-Mottaleb, M., Jain, A.K.: Face detection in color images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 24(5), 696–706 (2002)
14. Salvador, S., Chan, P.: FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. In: KDD Workshop on Mining Temporal and Sequential Data, pp. 70–80 (2004)
15. Hazen, T.J., Saenko, K., La, C.-H., Glass, J.R.: A segment-based audio-visual speech recognizer: Data collection, development and initial experiments. In: Proceedings of the International Conference on Multimodal Interfaces, State College, Pennsylvania, pp. 235–242 (October 2004)

16. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition* 29, 51–59.17. Cootes, T.F., Edwards, G.J., Taylor, C.J. Active appearance models. *IEEE Trans. On Pattern Recognition and Machine Intelligence* 23(6), 681–685 (1996)
17. Huang, X., Li, S.Z., Wang, Y.: Shape localization based on statistical method using extended local binary, *Image and Graphics*. In: *Proceedings. Third International Conference on Volume, Issue, December 18-20, 2004*, pp. 184–187 (2004)



# Continuous Stereo Gesture Recognition with Multi-layered Silhouette Templates and Support Vector Machines\*

Rafael Muñoz-Salinas<sup>1</sup>, Eugenio Aguirre<sup>2</sup>, Miguel García-Silvente<sup>2</sup>,  
and Moises Gómez<sup>2</sup>

<sup>1</sup> Department of Computing and Numerical Analysis.

Escuela Politécnica Superior

University of Córdoba, 14071 Córdoba, Spain

rmsalinas@uco.es

<sup>2</sup> Department. de Computer Science and Artificial Intelligence.

E.T.S. Ingeniería Informática University of Granada- 18071 Granada, Spain

{eaguirre,m.garcia-silvente}@decsai.ugr.es,moi\_gomez@hotmail.com

**Abstract.** This paper presents a novel approach for continuous gesture recognition using depth range sensors. Our approach can be seen as an extension of Motion Templates [1] using multiple layers that register the three-dimensional nature of the human gestures. Our Multi-Layered templates are created using *depth silhouettes*, the extension of binary silhouettes when depth information is available. Both the original Motion Templates and our extension have been tested using several classification approaches in order to determine the best one. These approaches include the use of Hu-moments (originally employed in [1]), PCA and Support Vector Machines. Finally, we propose a methodology for creating a continuous gesture recogniser using motion templates. The methodology is applied both to our representation approach and to the original proposal. In order to validate our proposal, several stereo-video sequences have been recorded showing eight people performing a total of ten different gestures that are prone to be confused when monocular vision is used. The conducted experiments show that our proposal performs a 20% better than the original method.

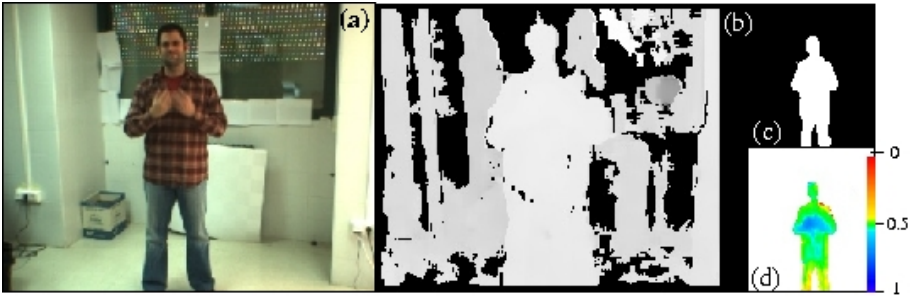
## 1 Introduction

Gesture recognition has become an important research area with many applications in the field of human-machine interaction. A great part of the effort has been focused on hand pose recognition and trajectory estimation since they are expressive means to communicate non-verbal information [3,7]. However, the analysis of the hands is not sufficient in many cases for a full description of the human movement. In areas like activity, pose and gait recognition, a more complete representation of the human body is required in order to characterise the

---

\* This work has been partially supported by the Spanish MEC project TIN2006-05565 and Andalusian Regional Government project TIC1670.

movement properly. In that sense, binary silhouettes are a representation mechanism commonly employed for these tasks [6,8]. Despite binary silhouettes are able to represent a wide variety of body configurations, using them for recognising complex activities is limited by several facts. First, gestures performed in front of the person torso are “invisible” in binary silhouettes (see Fig. 1(c)). Second, for some gestures, depth information is crucial for a proper recognition, e.g., think of the *point forward* and *point backward* gestures. Depth range information has been barely explored for gesture recognition despite of it is able to overcome some of the difficulties associated to monocular vision. In some works [9], people is represented by articulated body models whose joints are extracted from a point cloud generated using stereo vision. The main difficulty of that approach consists in detecting the joints that comprise the articulated body model and then tracking them avoiding the drifting problem. In [15], motion templates are extended to manage volumetric information gathered by a set of cameras surrounding the subject. Other authors have employed simpler approaches as reducing the number of tracked points or using colour information for a better tracking. For example, Nam *et al.* [11], present a system for gesture recognition based on tracking people hands. In the work of Yang *et al.* [16], face, hands and feet are tracked and their trajectories employed for gesture recognition. In [12], stereo vision is employed to determine the position where a user points to by combining hand tracking and face pose estimation. However, reducing the number of features generally leads to a reduction of the recognition capabilities of the method developed. Instead of tracking feature points, this work proposes a gesture recognition approach for depth range sensors based on the use of *depth silhouettes*, the natural extension of binary silhouettes when depth information is employed. In contrast to binary silhouettes, depth silhouettes are capable of registering movements performed in front of the person torso and only a pair of cameras are required to generate them. Furthermore, complex movements of the whole body can be represented without requiring the track of feature points. Our approach extends the work of Bobick *et al.* [1] to allow the use of depth information. In their work, they propose a representation model for gesture recognition, namely motion templates. In fact, two different types of templates are defined arguing that their combination obtains higher recognition rates. They are Motion History Images (MHIs), registering how the movement is performed, and Motion Energy Images (MEIs), focusing on where the movement is performed. Besides, they propose a gesture classification approach based on measuring the Mahalanobis distance to the seven Hu-moments [2]. In this work, motion templates are extended by the definition of Multi-Layered Silhouette Templates using information from depth silhouettes (instead of motion information). Moreover, we perform a study of three different recognition techniques using both the Bobick’s *et al.* method and ours. First, we analyse the performance of the classification method proposed in [1], based on measuring the Mahalanobis distance of the Hu-moments precomputed for the gestures learnt. Second, instead of using the Mahalanobis distance, Support Vector Machines (SVMs) are employed for learning the different gestures using the Hu-moments as feature vectors. Finally,



**Fig. 1.** (a) Image captured (b) Depth map (c) Binary silhouette (d) Depth silhouette

instead of using the Hu-moments, we test the combined use of Principal Component Analysis (PCA) and SVMs for gesture recognition in both representations. For experimentation purposes, we have employed a database of ten different gestures that are prone to be confused when using monocular vision. The results obtained show that the method proposed in this work outperforms the one based on monocular vision. Finally, we propose a methodology for creating continuous gesture recognisers by the model proposed in this work.

The remainder of this paper is structured as follows. Section 2 shows how depth silhouettes are generated. Section 3 explains the representation approach of 1 and our extension for depth range information. Section 4 test the three different recognition techniques using both Motion templates and Multi-Layered Silhouette Templates. Then, in Sect. 5, it is explained a methodology for creating a continuous gesture classifier employing our approach. Finally, Sect. 6 exposes some conclusions.

## 2 Silhouette Generation

Let us assume that the location of the person whose gestures are to be analysed is known and that his/her binary silhouette is being extracted and aligned. For that purpose, stereo detection and tracking algorithms developed in previous works [13, 14] have been employed in this work. Figure 1(a) shows an image captured with our stereo system and Figure 1(c) shows the resulting binary silhouette. White pixels of the binary silhouette represent foreground pixels in the original image and black pixels represent background information.

Our aim is to distinguish among a set of  $N_g$  different gestures, let us denote them by  $\{\Omega_1, \dots, \Omega_{N_g}\}$ . When information from a monocular system is used, a complete gesture is comprised by a variable number of consecutive binary silhouettes (let us denote them by  $\Theta^b = \{s_1^b, \dots, s_N^b\}$ ). The main limitation of binary silhouettes is that they are unable to represent gestures performed in front of the person's torso. Imagine the case of a person moving his/her hands in front of his/her torso (see Fig. 1(a)). In that case, hands are "hidden" in the binary silhouette, fused with the person's torso (Fig. 1(c)).

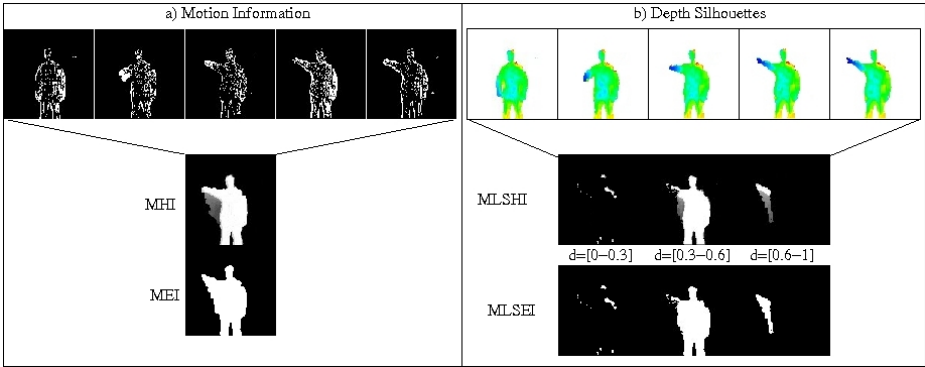
We define  $\Theta^d$  as the set of *depth silhouettes* representing a gesture taking into account depth range sensors. A depth silhouette is created from the binary one by assigning to every foreground pixel its corresponding depth value extracted from a depth map (see Fig. 1(b)). In our case, the depth map is created using stereo vision. As depth maps might have undefined values for some pixels, missing depth values are interpolated using neighbourhood information. Afterwards, the depth values of the silhouette are normalised to the range  $[0, 1]$  in the following way. Values in the range  $[h + \phi, h - \phi]$  are linearly translated to the range  $(0, 1]$ . The parameter  $h$  represents the depth of the person's mass centre (corresponding to the head and torso) and the parameter  $\phi$  is a depth limiting range around  $h$ . The limiting range  $\phi$  is such that includes the whole person with his/her arms extended. We have estimated empirically that  $\phi = 1$  meter is an appropriate value for the parameter. Finally, the value 0 is reserved for the pixels that are outside the limits of the range  $[h + \phi, h - \phi]$  and also for background pixels of the binary silhouette. Figure 1(d) shows the depth silhouette obtained from the depth map in Fig. 1(b) and the binary silhouette in Fig. 1(c). In Fig. 1(b), clearer pixels represent nearer points and black pixels represent undefined depth values. In Fig. 1(e), white pixels represent zero valued pixels. As it can be noticed, depth silhouettes overcome the limitation of binary silhouettes previously indicated. In depth silhouettes, a hand in front of the torso is registered as a mass of points with higher values than 0.5.

### 3 Multi-layered Silhouette Templates

This section explains the basis of Motion History Images (MHIs) and Motion Energy Images (MEIs) [1], and the extensions proposed in this work. A MHI,  $\mathcal{H}_\tau$ , is a compact representation for a sequence of silhouettes. In a MHI, the value of a pixel is a function of the temporal history of motion at that point and is calculated as:

$$\mathcal{H}_\tau(x, y, t) = \begin{cases} \tau & \text{if } D(x, y, t) = 1 \\ \max(0, \mathcal{H}_\tau(x, y, t - 1) - \Delta_\tau) & \text{otherwise,} \end{cases} \quad (1)$$

In Eq. 1,  $D(x, y, t)$  is the motion image obtained by image differencing,  $\tau$  is a temporal value indicating the duration of the movements to be recognised and  $\Delta_\tau = 1/\tau$ . The result of Eq. 1 is an intensity image, where brighter pixels represent regions with recent occupancy. As the time passes, older information reduces its intensity in the MHI thus becoming darker pixels. Fig. 2(a) show five of the fifteen silhouettes that form an execution of the *point forward* gesture. Below, there have been shown the corresponding MHI. As it can be noticed, the motion images show movement not only in the hand regions but also movement in the interior of the silhouette due to small movements of the subject while the gesture is performed. Thus, the results of that method are fairly identical if  $D(x, y, t)$  is substituted by silhouette information  $s_i^b(x, y)$  (as we have already checked). An MHI gives information about *how* a movement is performed. In contrast to a MHI, a MEI is employed to define *where* the movement is performed. A MEI,



**Fig. 2.** (a) Binary silhouettes for an execution of the *point forward* gesture. Below are the corresponding MHI and MEI. (b) Depth silhouettes for the *point forward* gesture. Below are the corresponding *MLSHI* and *MLSEI*. It can be noticed that the representation approach proposed (b) allows to register the three-dimensional nature of the gesture.

$\mathcal{E}_\tau$ , is a binary image consisting in the aggregation of all the silhouettes in a temporal window and can be obtained by thresholding the MHI as in Eq. 2. In Fig. 2(a) is shown the MEI of the *point forward* gesture.

$$\mathcal{E}_\tau(x, y, t) = \begin{cases} 1 & \text{if } \mathcal{H}_\tau(x, y, t) \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In this work, both motion templates are extended to manage depth information but instead of motion information, we employ the information of the depth silhouette to define the so called Multi-Layered Silhouette History Image *MLSHI* and the Multi-Layered Silhouette Energy Image *MLSEI*. A *MLSHI*,  $3\mathcal{H}_\tau$ , consists in multiple parallels MHI, each one registering the movement performed in a particular region of the space defined by a depth silhouette. Similarly, a *MLSEI* ( $3\mathcal{E}_\tau$ ) registers where the movement is performed. Let us denote by

$$3\mathcal{H}_\tau = \{3\mathcal{H}_\tau^1, \dots, 3\mathcal{H}_\tau^N\}, 3\mathcal{E}_\tau = \{3\mathcal{E}_\tau^1, \dots, 3\mathcal{E}_\tau^N\}$$

to the set of  $N$  layers that comprise a *MLSHI* and a *MLSEI*, respectively. Since depth silhouettes are normalised in the range  $[0, 1]$ , the  $3\mathcal{H}_\tau^i$  layer registers the movement in the depth range  $(\frac{i-1}{N}, \frac{i}{N}]$ . Then,  $3\mathcal{H}_\tau^i$  is calculated as:

$$3\mathcal{H}_\tau^i(x, y, t) = \begin{cases} \tau & \text{if } s_t^b(x, y) = 1 \text{ and } s_t^d(x, y) \in (\frac{i-1}{N}, \frac{i}{N}] \\ \max(0, 3\mathcal{H}_\tau^i(x, y, t-1) - \Delta_\tau) & \text{otherwise.} \end{cases} \tag{3}$$

As in the previous case, the *MLSEI* is obtained by thresholding the *MLSHI*, (as in Eq. 2). Figure 2(b) shows, the depth silhouettes for the same gesture shown in Figure 2(a). As it can be seen, the fact that the person moves the arm in front of him is properly represented in the depth silhouette (notice the blue colour of the arm when it is extended). The corresponding *MLSHI* and *MLSEI* (using  $N = 3$  layers) are shown below in the figure.

## 4 Evaluation of Several Gesture Recognition Approaches

This section performs an evaluation of several approaches for gesture recognition using the representation models explained in the previous section. First, we explain the basis of the original approach presented in [1]. Then, we introduce the use of PCA for data reduction in order to analyse whether it improves the recognition performance.

### 4.1 Hu-Moment-Based Classification

The first classification approach examined is the one proposed by Bobick *et al.* [1] for gesture recognition with binary silhouettes. That approach is based on the use of the seven Hu-moments [2] which have been used for scale, position, and rotation invariant pattern identification. They are defined as:

$$\begin{aligned}
 h_1 &= \eta_{20} + \eta_{02} ; h_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
 h_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 ; h_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
 h_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
 &\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
 h_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
 h_7 &= (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] - \\
 &\quad (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
 \end{aligned} \tag{4}$$

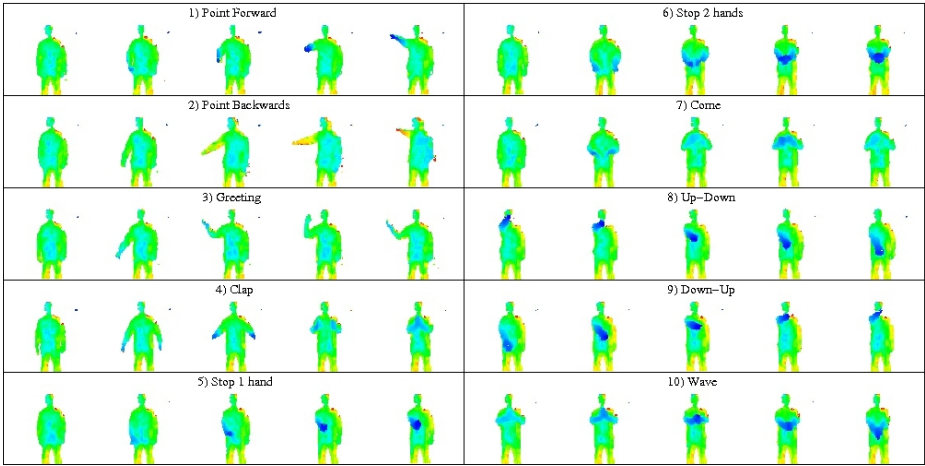
where  $\eta_{ij}$  are normalized central moments of 2-nd and 3-rd orders. In [1], the seven moments are calculated for all the patterns and the mean and covariance matrices are calculated for each class. Then, for an unseen pattern, the moments are calculated and also the Mahalanobis distance to every class. The unseen pattern is assumed to belong to the nearer class. This method is easily extended to our multi-layered approach by aggregating the seven moments for each layer into a single feature vector. Finally, Hu-moments have also been tested in combination with SVMs [4] in order to test whether both classification approaches differ in the results obtained.

### 4.2 Data Reduction Using PCA

The analysis of principal components have been widely employed for data reduction [5]. In this work, PCA has been employed in order to reduce the templates to its principal components. Our goal is to determine whether PCA outperforms Hu-moments in representing templates. Given the set of all the templates generated for the gestures in our database using any of the representation method previously explained (MHI, MEI, MLSHI or MLSEI), we form a set of vectors  $\{x^t\}$ , where  $x \in \mathcal{R}^{N=mn}$ , by lexicographic ordering of the pixel elements of the templates. Then, the eigenvalue problem is defined as

$$\Lambda = \Phi^T \Sigma \Phi$$

where  $\Sigma$  represents the covariance matrix,  $\Phi$  is the eigenvector matrix of  $\Sigma$ , and  $\Lambda$  is the diagonal matrix of eigenvalues. Once the eigenvalue problem is solved,



**Fig. 3.** Ten gestures employed for testing our recognition approach. There are shown the five most representative depth silhouettes of each gesture.

using for example Singular Value Decomposition, a template can be transformed into its principal component feature vector  $y = \Phi_e^T \tilde{x}$ , where  $\tilde{x} = x - \bar{x}$  is the mean-normalised image vector and  $\Phi_e^T$  is a submatrix containing the  $e$  eigenvectors with higher eigenvalues. Thus, the resulting vector  $y$  has  $e$  elements and corresponds to a point in the orthogonal coordinate system defined by the eigenvectors. In particular, the subspace defined using PCA has the distinction of keeping the largest variance. In our multi-layered approach, the different layers are reduced independently and then aggregated into a single feature vector.

### 4.3 Selection of the Best Classification Approach

In order to test our approach, the  $N_g = 10$  different gestures shown in Fig. 3 have been employed. Some of the gestures selected are difficult cases for monocular vision. In some of them, the movement is performed in front of the person’s torso. Others, like *point forward* and *point backward*, are prone to be confused when using monocular vision. A total of eight different people were instructed to perform each gesture fifteen times, thus obtaining a total of 120 instances of each gesture summing up 20062 silhouettes. The sequences were recorded using a *Bumblebee* stereo camera from the *Point Grey Research* manufacturer. The stereo system is comprised by two coplanar cameras separated by a distance of 12 cm and is able to record sequences of size  $320 \times 240$  at a frame rate of 15 fps. The stereo correspondence algorithm employed is an improved version of SAD (provided by the manufacturer) that performs subpixel interpolation. For training purposes, we have employed a *leave-one-person out* approach. That is to say, training is performed leaving apart all the gestures of a person. Then, the performance of the trained system is tested on the patterns of the “unseen”



person. The process is repeated for each person and the final success is obtained as the average value.

The three different recognition approaches previously explained have been tested. In all of them, the monocular approach presented in [1] and the multi-layered approach presented in this work have been tested. As previously indicated, in [1] is argued that the combined use of MEI and MHI increase the recognition rate. In order to determine the degree in which the results are improved, the combined and isolated use of the two type of templates are tested using both the monocular and depth approaches. The templates are combined by aggregating of the individual feature vectors of each template into a single one. The results obtained are shown in Table 1.

**Table 1.** Classification results for the different approaches tested

Method	MEI(%)	MHI(%)	M(E+H)I(%)	MLSEI(%)	MLSHI(%)	MLS(E+H)I(%)
Hu+Mahalanbs	25	31	37	33	31	10
Hu+SVM	35	12	36	36	13	39
PCA+SVM	54	68	68	76	86	<b>88</b>

Each row of the table represents the percentage of success for a different recognition approach, employing both Motion Templates and Multi-Layered Silhouette Templates. The first column indicates the method employed. The second, third and fourth columns represent the results for the MEI, MHI and the combined use of MEI and MHI, respectively. Finally, the last three columns present the results for the multi-layer approach. The last row shows the results when PCA is used. In that case, the training process was repeated for a different number of principal components although the table shows only the results for the best case.

At the light of the results obtained, it can be noticed that the multi-layered approach presented in this work outperforms the representation approach based on binary silhouettes. While Motion Templates obtains a success of 68% in the best case, the best multi-layered approach obtains a success of 88%. In regards to the combined use of MEI and MEI, it can be seen that while in the monocular approach the results are improved, in our approach it is not always true for our approach.

## 5 Continuous Gesture Recognition

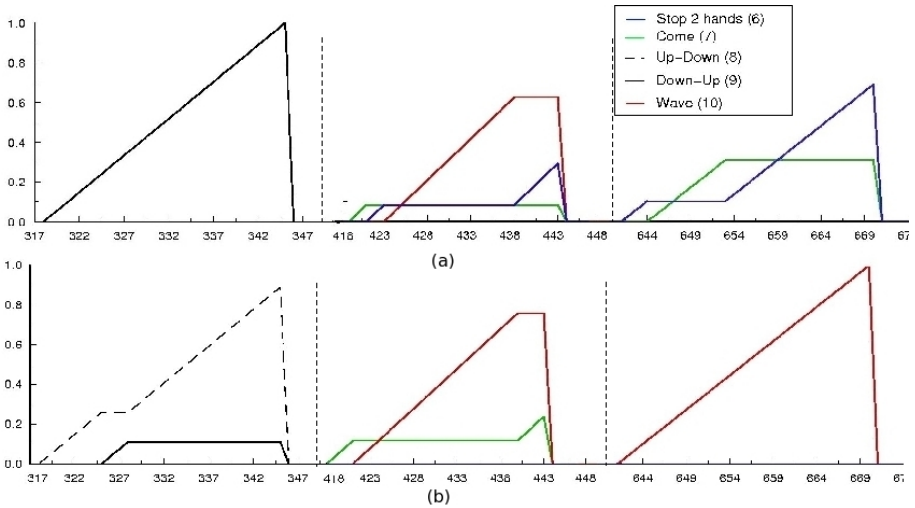
The approach explained up to this moment can be employed to create a continuous recogniser. However, an important aspect that must be taken into account is to distinguish the gestures learnt from involuntary movements or from the execution of other gestures that have not be learnt. Our approach for continuous recognition consists in employing two different SVM classifiers. The first one is a binary classifier that detects the presence or absence of some of the gestures learnt and is employed at each frame in order to detect whether a gesture is being





**Fig. 4.** Images from a sequence employed for testing the continuous recogniser. Gestures up-down, wave, and *stop 2 hands* are shown

performed. The second classifier is activated whenever the first one detects the presence of a gesture for which is trained. For that purpose, the best recogniser of the previous section has been employed. The first classifier has been trained as follows. There have been recorded several stereo-video sequences in which users were instructed to perform the gestures shown in Fig. 3 in a random order. Afterwards, the sequences were examined by a human in order to determine the start and end frames for each gesture. Then, the motion templates generated between the start and end of a gesture have been employed to train a one-class SVM [10]. The goal is that the first classifier detects the presence of a gesture during the frames in which it is being performed (let us call this period of time *temporal gesture window* (*tgw*)), and during this time the second classifier is activated to detect the gesture. Nevertheless, there is a problem that must be solved. It must be noticed that during a *tgw*, the second classifier might produce different (and some of them erroneous) outputs. To avoid that problem, the detection of a gesture is done by a voting process such as the most voted gesture at the end of a *tgw* is assumed to be valid one. Figures 4 and 5 show the results of the continuous classifier for one of the video tested sequences. For testing purposes, two different classifiers have been created. One using our multi-layered method, and another one using the original templates. Figure 4 presents some selected frames corresponding to the execution of three different gestures. Frames [317 – 347] show the execution of the *up-down* gesture. In frames [418 – 443] the person executes a *wave* gesture, and frames [640 – 670] show the execution of the *stop 2 hands* gesture. The recognition results of our approach and the original one are presented in Fig. 5(a) and Fig. 5(b), respectively. Each graph shows as a different coloured line the evolution of the number of votes (normalised in the range [0, 1]) that each gesture receives during the corresponding *tgw*. As it can be noticed, there are differences between the results obtained by the two methods. For the frames [317 – 347], our classifier identifies correctly the execution of the *up-down* gesture. However, the recogniser based on binary silhouettes indicates first that the gesture is *down-up*, then, *up-down* and finally, *down-up* again. At the end, *down-up* is the most voted gesture so that the real gesture is incorrectly classified. Then, in frames [418 – 443], our recogniser indicates the execution of *come*,



**Fig. 5.** (a) Results obtained using our Multi-Layered approach. (b) Results obtained using the original Motion Templates.

*stop 2 hands* and *wave*, but, as it can be seen, *wave* is the gesture that obtains the majority of the votes. Thus, the gesture is correctly recognised. In the frames [418 – 443], the approach based on binary silhouettes is also able to recognise the gesture appropriately. Finally, for the frames [640 – 670], our approach identifies correctly the execution of the gesture while the original approach fails.

## 6 Conclusions

This paper has presented a novel approach for continuous gesture recognition using depth range information. Our approach, Multi-Layered Silhouette Templates, can be seen as an extension of Motion Templates [1] in order to manage depth information. Multi-Layered Silhouette Templates are created using *depth silhouettes*, the extension of binary silhouettes when depth information is available. Both our representation model and [1] are tested using three different recognition techniques in order to select the most appropriated one. First, the performance of the technique proposed in [1] is examined. It is based on measuring the Mahalanobis distance of the Hu-moments. As second approach, we have examined the combined use of Hu-moments with Support Vector Machines (SVMs). Finally, we have tested the use of Principal Component Analysis (PCA) and SVMs. The results obtained indicate that: a) the combined use of PCA and SVMs outperforms the other two techniques; and b) our Multi-Layered representation outperforms the original proposal [1].

Finally, a methodology for continuous gesture recognition using SVMs is presented. It is based on the use of two classifiers. A first one that identifies when gestures are being performed and a second one that detects which is the ges-

ture by a voting process. The performed experiments show that our approach outperforms the original one also in continuous recognition.

## References

1. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 257–267 (2001)
2. Borenstein, J., Koren, Y.: Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory* 8, 179–187 (1962)
3. Chen, F.S., Fu, C.M., Huang, C.L.: Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and Vision Computing* 21, 745–758 (2003)
4. Cortes, C., Vapnik, V.: Support-vector network. *Machine Learning* 20, 273–297 (1995)
5. Fukunaga, K.: *Introduction to Statistical Pattern Recognition*. Academic Press, London (1990)
6. Kale, A., Sundaresan, A., Rajagopalan, A., Cuntoor, N., RoyChowdhury, A., Kruger, V., Chellappa, R.: Identification of humans using gait. *IEEE Transactions on Image Processing* 13, 1163–1173 (2004)
7. Kang, H., Lee, C.W., Jung, K.: Recognition-based gesture spotting in video games. *Pattern Recognition Letters* 25, 1701–1714 (2004)
8. Lam, T.H.W., Lee, R.S.T., Zhang, D.: Human gait recognition by the fusion of motion and static spatio-temporal templates. *Pattern Recognition* 40, 2563–2573 (2007)
9. Luck, J., Small, D., Little, C.Q.: Real-time tracking of articulated human models using a 3d shape-from-silhouette method. In: Klette, R., Peleg, S., Sommer, G. (eds.) *RobVis 2001*. LNCS, vol. 1998, pp. 19–26. Springer, Heidelberg (2001)
10. Manevitz, L.M., Yousef, M.: One-class svms for document classification. *Journal of Machine Learning Research* 2, 139–154 (2001)
11. Nam, Y., Wahn, K.: Recognition of hand gestures with 3d, nonlinear arm movement. *Pattern Recognition Letters* 18, 105–113 (1997)
12. Nickel, K., Stiefelwagen, R.: Visual recognition of pointing gestures for human-robot interaction. *Image and Vision Computing* (in press, 2007)
13. Muñoz Salinas, R., Aguirre, E., García-Silvente, M.: People detection and tracking using stereo vision and color. *Image and Vision Computing* (25), 995–1007 (2007)
14. Muñoz Salinas, R., Aguirre, E., García-Silvente, M., González, A.: People detection and tracking through stereo vision for human-robot interaction. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) *MICAI 2005*. LNCS (LNAI), vol. 3789, pp. 337–346. Springer, Heidelberg (2005)
15. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding* 104, 249–257 (2006)
16. Yang, H-S., Kim, J-M., Park, S-K.: Three dimensional gesture recognition using modified matching algorithm. In: Wang, L., Chen, K., Ong, Y.S. (eds.) *ICNC 2005*. LNCS, vol. 3611, pp. 224–233. Springer, Heidelberg (2005)

# Small-Time Local Controllability of a Differential Drive Robot with a Limited Sensor for Landmark-Based Navigation

Rafael Murrieta-Cid and Jean-Bernard Hayet

Centro de Investigación en Matemáticas, CIMAT  
Guanajuato, México  
{murrieta,jbhayet}@cimat.mx

**Abstract.** This work studies the interaction of the nonholonomic and visibility constraints of a robot to maintain visibility of a landmark. The robot is a differential drive system (nonholonomic robot) and has a sensor with limited capabilities (limited field of view). In this research, we want to determine whether or not a robot can always maintain visibility of a landmark during the execution of a path between any two locations. We present two kinematic models. First, a robot with 3 controls, where the controls correspond to the two wheels velocities plus one independent controlled sensor. Second, a model with only 2 controls, which controls both the wheels and the sensor rotation. We show that our system (with 3 or 2 controls) is small-time local controllable.

## 1 Introduction

Landmarks have been frequently used in robotics, either to localize the robot with respect to them [2,21] or to navigate [14,17,4,8]. In robot navigation, a landmark can be used as a goal or sub-goal that the robot must reach or perceive during the robot motion. A landmark can be defined in several manners: From a set of points in an image having useful properties (i.e. saliency), up to a 3 D object associated with a semantic label and having 3-D position accuracy.

To use landmarks in the context of mobile robotics, the first basic requirement is to perceive them during the robot motion. Even though, landmarks have been extensively used in robotics [5,2,21,14,4], to our knowledge *this is the first attempt* to show whether or not a robot under the interaction of its nonholonomic and visibility constraints is small time local controllable. We believe that our research is very pertinent given that a lot of mobile robots are nonholonomic systems equipped with limited field of view sensors (lasers or cameras).

Our final goal is to show that if a collision free path between any two locations exists for an unconstrained system (i.e. a holonomic robot equipped with omnidirectional field of view sensor) to maintain landmark visibility, then a feasible path shall exist for our system under their motion and perception constraints.

In other words, we want to determine whether or not a constrained robot can always maintain visibility of a landmark during the execution of a path between any two locations. To do so, we need to determine whether or not the system is small time locally controllable. This is the main motivation of this work.

Small time local controllability (STLC) implies that a system can locally maneuver in any direction. Let us define  $\mathcal{M}$  as a manifold; if the system is STLC at all  $x \in \mathcal{M}$ , then the system can follow any curve on  $\mathcal{M}$  arbitrarily closely. This has an important implication in environments with obstacles or sensor constraints, as a STLC system can always maneuver through clutter space, since any motion of a system with no motion constraints can be approximated by a motion constrained system that is STLC anywhere [11,6].

In [3] we have shown that a differential drive robot (DDR) with field-of-view and range constraints is controllable. We have also provided the shortest paths in the sense of distance for the field-of-view constraint (without the range constraint). However, in that work we have not shown whether or not the system is small time locally controllable. STLC implies controllability but the converse property is not true. The difference between controllability and STLC can be described with an intuitive example (a definition is given in section 2.1), a car that can only go forward (Dubins car [7]) is controllable, but it is not small time local controllable. Such a car will need a long motion to reach a close position behind it. Note that such a long motion may produce robot collision (in an environment with obstacles) or force the robot to lose landmark visibility by moving outside the sensor limits.

The research reported in this paper differs from our previous efforts in the following main point: We shall present an existence proof of small time-local controllability of the system under its motion and perception constraints. We will present two robot models. First, a robot with 3 controls, where the controls correspond to the two wheel velocities plus one independent controlled sensor. Second, a robot model with 2 controls. In this last model, the sensor is a slave of the robot's wheels, thus, the sensor moves in function of the wheels' motors. The model with 2 controls assumes that the robot is moving along the optimal paths (composed by straight lines and sectors of spirals) presented in [3], as the visibility constraint is satisfied exactly at each point.

*Problem definition:* The initial and final locations and the path can be given to the robot as input or be computed by a planner. The initial and final configurations satisfy the visibility constraint. The robot is a differential drive system (nonholonomic robot). The robot is equipped with a limited field of view sensor. The landmark is static and the robot observer must maintain the landmark within the sensor's field of view. Visibility is defined geometrically, i.e. the landmark is visible from the robot sensor if a clear line of sight can join them and the landmark is within the minimal and maximal sensor field of view. Thus, this work focuses on studying the interaction of the nonholonomic and visibility constraints of the robot observer. More precisely, the problem to solve here consists in proving whether or not this system is small time locally controllable under its kinematic and perceptual constraints.

## 2 Nonholonomic Constraints

Nonholonomic systems are characterized by constraint equations involving the time derivatives of the system configuration variables. In a nonholonomic system these equations are not integrable; in fact this property corresponds to the core of the definition of a nonholonomic system. The state transition equation of a system  $\dot{X} = f(X, U)$ , where  $X$  is the state vector, indicates how the state of the system changes over the time according to some inputs. If the state transition equation is integrable, it is said that the corresponding system is holonomic; otherwise, the system is nonholonomic (Frobenius theorem, see below).

Typically, non-integrable equations arise when the system has less controls than configuration variables. For instance, a differential drive robot has two controls (right and left wheel velocities) while moving in a 3-dimensional configuration space  $X = (x, y, \theta)^T$ .

From the point of view of motion planning, the main implication of nonholonomic constraints is that any collision-free path in the configuration space does not necessarily correspond to a feasible path for the system. Purely geometric techniques to find collision-free paths do not apply directly to nonholonomic systems.

Motion planning with nonholonomic constraints has been a very active research field (a nice overview is given in [12]). The most important results in this field have been obtained by addressing the problem with tools from differential geometry and control theory. Laumond pioneered this research [9] and produced the result that a free path for a free-flying robot moving among obstacles in a 2d work-space can always be transformed into a free path for a nonholonomic car-like robot by making car maneuvers [11]. Sussman and Liu [20] propose an algorithm for constructing a sequence of admissible trajectories in the presence of obstacles for nonholonomic systems. Li and Canny [15] apply controllability theory for non-linear systems to nonholonomic robots. Murray and Sastry [16] use sinusoidal control in steering nonholonomic systems.

The study of optimal paths for nonholonomic systems has also been an active research topic. Dubins [7] determined the shortest paths for a car-like robot than can only go forward. Reed and Shepp extended this work and established the shortest length paths for a car-like robot that can move forward and backward [19]. In [18] a complete characterization of the shortest path for a car-like robot is given.

Balkcom and Mason determined the time-optimal trajectories for a differential drive robot [1]. These trajectories for a differential drive robot were found using Pontryagin's Maximum Principle and geometric analysis. All these results about optimal paths assume that the nonholonomic robot moves in the free space (without obstacles).

### 2.1 Integrability and Controllability

The Frobenius and Chow theorems are used to establish integrability and controllability.

A system is said to be nonholonomic if the state transition equation is not integrable. *Frobenius theorem establishes that the state transition equation is integrable if and only if all the vectors fields that can be obtained by the Lie bracket operations are contained in  $\Delta$ .*

$\Delta$  is called distribution and can be considered as a vector space.  $\Delta = \text{span}(\alpha^1(x), \alpha^2(x), \dots, \alpha^n(x))$  is the set of vector fields  $\alpha^i$ ,  $\alpha^i(x)$  is a vector-valued function of  $x$ .

Two important properties to be determined in a nonholonomic system are controllability and small time local controllability (STLC).

Controllability means that the nonholonomic robot is able to overcome its differential constraints by using velocities that are not directly permitted by the state transition equation. That implies that any state can be reached from any other state.

If a system is STLC then the system can move in any direction, in an arbitrarily small amount of time. More formally, let  $R(x, \delta t)$  denotes the set of all points reachable in time  $\delta t$ . A system is small-time locally controllable if for all  $x \in X$  and any  $\delta t > 0$ ,  $x$  lies within an open set contained by  $R(x, \delta t)$  [13]. These new velocities not directly permitted in the state transition equation are generated with the Lie Bracket.

The  $i^{\text{th}}$  component of the Lie bracket is given by:

$$Z_i = \sum_{j=1}^n (P_j \frac{\partial Q_i}{\partial x_j} - Q_j \frac{\partial P_i}{\partial x_j}), \quad (1)$$

where  $P_j$  and  $Q_j$  are vector fields.

Chow's theorem says that *a system is small-time controllable if and only if the dimension of  $CLA(\Delta)$  is the dimension of  $X$  and the system is symmetric* [6], where  $CLA$  is the Control Lie Algebra,  $CLA(\Delta)$  is the set of of vector fields  $\alpha^i$  and all the new linearly-independent vectors that can be generated from the Lie bracket operations by using the control inputs.  $X$  is the robot state (velocities that can be applied to the configuration space variables). Note that the dimension of  $CLA(\Delta)$  can never be greater than the dimension of robot state  $X$ .

### 3 Visibility Constraints

In the problem addressed in this work, a differential drive robot must maintain visibility of a static landmark. The robot is equipped with a sensor that has a limited field of view. It is assumed that the robot is equipped with a pan controllable sensor.

We make the usual assignment of body attached frame to the robot. The origin is at the midpoint between the two wheels,  $y$ -axis parallel to the axle, and the  $x$ -axis pointing forward, parallel to the robot's heading.  $\theta$  is the angle from the world  $x$ -axis to the robot  $x$ -axis. The robot can move forward and backward,

the heading of the robot is defined as the direction in which the robots moves, so the heading angle with respect to the robot's  $x$ -axis is zero or  $\pi$ .

The Landmark is at the origin of the coordinate system. The position of the robot with respect to the landmark is defined in polar coordinates by the following equations:

$$r = \sqrt{x^2 + y^2} \quad (2)$$

$$\alpha = \arctan \frac{y}{x} \quad (3)$$

The sensor is placed so that the optical center lies directly above the origin of the robot's local coordinate frame. The sensor pan angle  $\phi$  is the angle from the robot's  $x$ -axis to the optical axis. The range of the sensor rotation is limited, such that  $\phi \in [\phi_1, \phi_2]$ . Figures 1 and 2 show these conventions.

## 4 Proving that the System is Small Time Local Controllable

We present two kinematic models. First, a robot with 3 controls, where the controls correspond to the two wheel velocities plus one independent controlled sensor. Second, a model with only 2 controls, where those inputs control both the wheels and the sensor rotation.

### 4.1 Differential Drive Robot with 3 Controls

In this case the system constraints are the following:

1. The system is a differential drive robot (nonholonomic system) that can move forward and backward, hence it is a symmetric system.
2. The landmark is always within the sensor field of view, i.e.,

$$(\alpha + \pi) - \theta - \epsilon \leq \phi \leq (\alpha + \pi) - \theta + \epsilon \quad (4)$$

where  $\epsilon > 0$  represents the sensor half field of view (See figure 1).

3. The sensor pan angle is limited such that  $\phi \in [\phi_1, \phi_2]$ . The sensor is pointed forward  $0 \in [\phi_1, \phi_2]$ , and hence the robot can directly approach the landmark.

The robot configuration is given by  $X = (x, y, \theta, \phi)$  or equivalently by polar coordinates  $(r, \alpha, \theta, \phi)$ . Note that  $\phi$  defines the sensor orientation with respect to the robot heading; the sensor does not need to be directly pointing to the landmark, but the landmark must be within the sensor's field of view. This setting gives flexibility with respect to sensor heading and hence is tolerant to control errors. However, to define this model 4 configuration variables and 3 controls are needed.



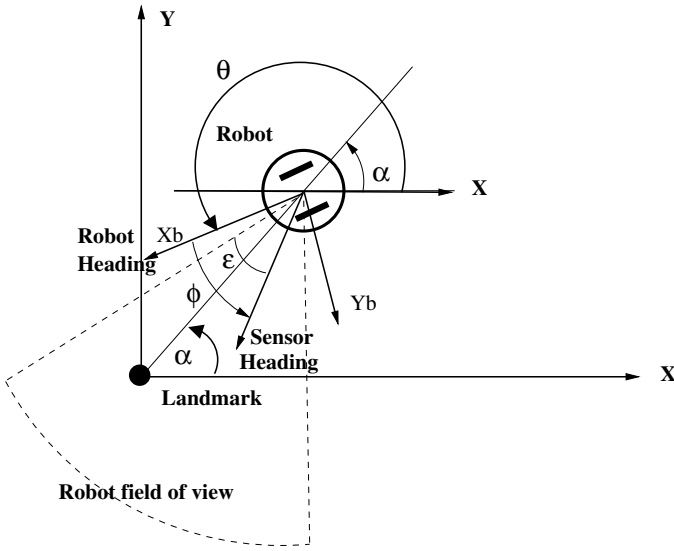


Fig. 1. Differential drive robot with 3 controls

In this case, the state transition equation for our differential drive robot equipped with a controllable sensor can be expressed as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 & 0 \\ \sin \theta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} \quad (5)$$

The controls of our system are  $\mu_1$ ,  $\mu_2$  and  $\mu_3$ . Controls  $\mu_1$ ,  $\mu_2$  are the same than in the differential drive robot without the sensor, where  $\mu_1 = w_r + w_l$  and  $\mu_2 = w_r - w_l$ .  $w_r$  and  $w_l$  are the angular velocities of the left and right wheels respectively. Control  $\mu_1$  means going in a straight-line motion and control  $\mu_2$  corresponds to a rotation centered on the robot.

A new control must be added to the system given that the sensor is moved by an independent motor. This new control is  $\mu_3 = w_c$ , where  $w_c$  is the sensor angular velocity. Control  $\mu_3 = 1$  corresponds to a sensor rotation.

Chow's theorem is used in order to prove that the system is STLC (see section 2). The first step to know the dimension of  $CLA(\Delta)$  is to compute the Lie brackets of all the original vectors fields. The original vector fields are:  $\vec{V} = [\cos \theta \ \sin \theta \ 0 \ 0]$ ,  $\vec{W} = [0 \ 0 \ 1 \ 0]$  and  $\vec{X} = [0 \ 0 \ 0 \ 1]$ . The dimension of the distribution  $\Delta$  of this system is 3 and the dimension of state  $X$  is 4.

The possible vector fields combinations are:  $[\vec{V}, \vec{W}]$ ,  $[\vec{V}, \vec{X}]$ ,  $[\vec{W}, \vec{X}]$ . The resulting components of the Lie bracket for the different vector field combinations are:  $\vec{Z} = [\vec{V}, \vec{W}] = [\sin \theta \ -\cos \theta \ 0 \ 0]$ , Lie bracket for the other combinations are null vectors.

$\vec{Z}$  is linearly independent from  $\vec{V}$ ,  $\vec{W}$  and  $\vec{X}$ . The determinant of the matrix A:

$$A = \begin{pmatrix} \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \sin \theta & -\cos \theta & 0 & 0 \end{pmatrix}; \quad \begin{vmatrix} \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \sin \theta & -\cos \theta & 0 & 0 \end{vmatrix} \neq 0 \quad (6)$$

is nonzero everywhere in the configuration space. Thus, the dimension of the  $CLA(\Delta)$  is equal to the dimension of the system state  $X$ . Both of them are 4. Therefore, the system is nonholonomic and small time local controllable.  $\square$

### 4.2 Differential Drive Robot with 2 Controls

It is also possible to generate a state transition equation with just two controls. In this scheme, the sensor is pointing to the landmark according to the wheels velocities, thus the sensor control is not needed. As before, it must also be determined, whether or not this system is STLTC.

In this case the constraints (1) and (3) are the same that in the case of a system with 3 controls. However, constraint (2) is stronger.

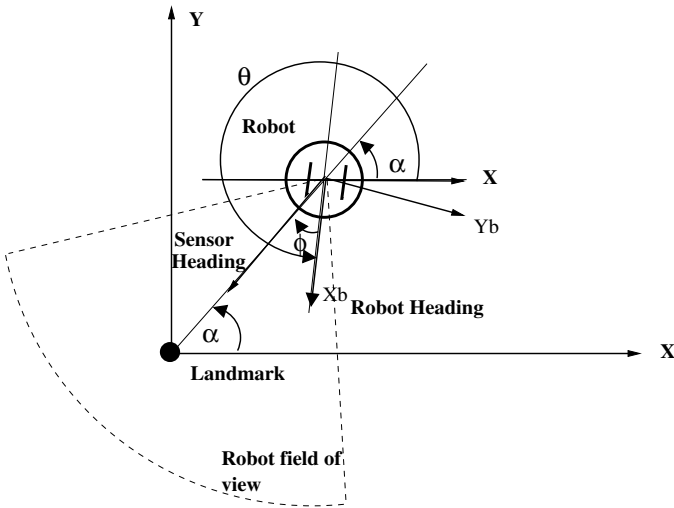


Fig. 2. Differential drive robot with 2 controls

The sensor optical axis is always pointing toward the landmark (see figure 2), so that :

$$\theta = \alpha - \phi + \pi. \quad (7)$$

The construction of the state transition equation is as follows. First, we get the derivative of equation 7:

$$\dot{\phi} = \dot{\alpha} - \dot{\theta}. \quad (8)$$

Then we obtain the derivative of equation [3](#),

$$\dot{\alpha} = \frac{1}{1 + \frac{y^2}{x^2}} \frac{\dot{y}x - \dot{x}y}{x^2} = \frac{\dot{y}x - \dot{x}y}{x^2 + y^2}. \tag{9}$$

The controls are:

$$\mu_1 = w_r + w_l \tag{10}$$

$$\mu_2 = w_r - w_l \tag{11}$$

Therefore, the

$$\dot{\theta} = w_r - w_l \tag{12}$$

$$\dot{x} = \cos \theta (w_r + w_l) \tag{13}$$

$$\dot{y} = \sin \theta (w_r + w_l) \tag{14}$$

The key observation is the following:  $\phi$  is not a degree of freedom any more. It can be expressed as a function of  $x, y$  and  $\theta$ . Hence, the robot configuration is totally defined by  $(x, y, \theta)$ .  $\phi$  and  $\dot{\phi}$  are constraints that the system must satisfy to maintain landmark visibility.

$\dot{\phi}$  can be expressed directly in function of the controls  $\mu_1, \mu_2$  and the configuration variables  $(\theta, x, y)$ . This can be done by substituting in [8](#), the values of  $\dot{\alpha}$  and  $\dot{\theta}$  from equations [9](#) and [12](#),

$$\dot{\phi} = \frac{(y \cos \theta - x \sin \theta) \mu_1}{x^2 + y^2} - \mu_2. \tag{15}$$

Hence, the state transition equation takes the form:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \tag{16}$$

which is exactly the same of the differential drive robot [11,13](#). Because of the visibility restrictions, the two only basic motions are straight lines and logarithmic spirals [3](#). We have already seen that the vector field associated to the straight line is  $(\cos \theta, \sin \theta, 0)^T$ . Now let us express the vector field associated to the spirals. The equations of these curves are [3](#):

$$r = r_0 e^{(\alpha_0 - \alpha) / \tan \phi} \tag{17}$$

where  $r_0, \alpha_0$  and  $\phi$  remain constant along these curves. From this equation, we can easily derive the corresponding vector field, as  $x = r \cos \alpha$  and  $y = r \sin \alpha$  :

$$\vec{W} = \begin{pmatrix} -y - \frac{x}{\tan \phi} \\ x - \frac{y}{\tan \phi} \\ 1 \end{pmatrix}. \tag{18}$$

Now remember that  $\phi$  is a constant on all spirals, but its value is constrained by the robot position, as  $\phi = \alpha - \theta + \pi$ . Moreover, as  $\alpha = \arctan \frac{y}{x}$ , we can write :

$$\frac{1}{\tan \phi} = \frac{1}{\tan(\arctan \frac{y}{x} + \pi - \theta)} = \frac{1 - \tan(\arctan \frac{y}{x}) \tan(\pi - \theta)}{\tan(\arctan \frac{y}{x}) + \tan(\pi - \theta)} = \frac{x + y \tan \theta}{y - x \tan \theta}.$$

Note that this vector field is not defined for  $y = x \tan \theta$ , which corresponds to zones where the robot has to follow a straight line, in fact, as it is pointing to the landmark. The last equation is combined with Eq.18 to get:

$$\vec{W} = \begin{pmatrix} -\frac{x^2+y^2}{y-x \tan \theta} \\ -\tan \theta \frac{x^2+y^2}{y-x \tan \theta} \\ 1 \end{pmatrix}. \tag{19}$$

The Lie bracket  $\vec{Z} = [\vec{V}, \vec{W}]$  can be computed by using formula of Eq.11 which leads to

$$\vec{Z} = \begin{pmatrix} \sin \theta - 2 \frac{x \cos \theta + y \sin \theta}{y-x \tan \theta} \\ -\cos \theta - 2 \tan \theta \frac{x \cos \theta + y \sin \theta}{y-x \tan \theta} \\ 0 \end{pmatrix}.$$

Last step of our demonstration, the determinant of the matrix  $B$ :

$$B = \begin{pmatrix} (\vec{V}, \vec{W}, \vec{Z}) \\ \cos \theta & -\frac{x^2+y^2}{y-x \tan \theta} & \sin \theta - 2 \frac{x \cos \theta + y \sin \theta}{y-x \tan \theta} \\ \sin \theta - \tan \theta \frac{x^2+y^2}{y-x \tan \theta} & -\cos \theta - 2 \tan \theta \frac{x \cos \theta + y \sin \theta}{y-x \tan \theta} & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{20}$$

is +1 for all values of  $x, y, \theta$ . Hence, again this system is small time local controllable everywhere in the configuration space.  $\square$

## 5 Conclusion and Future Work

It is possible to draw analogies between our system and the car-like robot. In the car-like robot the steering angle is constrained due to mechanical stops in the steering gear. This constraint forces the robot to move with curvature bounds [11]. In our system, these bounds also exist [3] due to the constraint of maintaining landmark visibility and the assumption that the sensor can rotate a limited angle.

In the car-like robot the curvature constraint is upper-bounded by the same value wherever it is defined [10]. In our system the bound changes according to the orientation of the robot with respect to the landmark.

In this paper, we have shown that our systems (with 3 or 2 controls) are small-time local controllable.

We are currently addressing the problem of landmark-based navigation in an environment with obstacles. Besides, the limitation in the sensor field of view induces the presence of a virtual obstacle in the configuration space. Therefore, it is necessary that there exists an open set around the robot to prove the existence of a path between any initial and final robot locations if one exists for an unconstrained system.

As in the case of a car-like system, the main difficulty in our system consists in determining the “size” of the free space allowed around the steering solution.

A way to compute this size is to determine the shortest paths for the system, this would allow us to determine a metric of the system. In [3] we have provided the shortest paths to maintain visibility of a landmark for a differential drive robot with field-of-view constraint. We plan to use those paths to determine lower and upper bounds of the paths’ metric and the metric’s induced topology.

The existence of obstacles (real or induced by sensor limitations) forces the use of some given metric in the plane. Given that the system is STLCL, if the shortest-path metric induces the same topology as the metric used to measure the distance between the robot and the obstacles, then the existence of any “small”  $\epsilon > 0$  clearance between the robot and the obstacles in the plane would be enough to guarantee a feasible path.

## Acknowledgments

This research was partially funded by CONACyT project 56754.

## References

1. Balkcom, D.J., Mason, M.T.: Time Optimal Trajectories for Differential Drive Vehicles. *International Journal of Robotics Research* 21(3), 199–217 (2002)
2. Betke, M., Gurfvits, L.: Mobile Robot Localization using Landmarks. *IEEE Transactions on Robotics and Automation* 13(2), 251–263 (1997)
3. Bhattacharya, S., Murrieta-Cid, R., Hutchinson, S.: Optimal Paths for Landmark-based Navigation by Differential Drive Vehicles with Field-of-View Constraints. *IEEE Transactions on Robotics* 23(1), 47–59 (2007)
4. Briggs, A.J., Detweiler, C., Scharstein, D., Vandenberg-Rodes, A.: Expected Shortest Paths for Landmark-Based Robot Navigation. *International Journal of Robotics Research* 12(8) (July 2004)
5. Bulata, H., Devy, M.: Incremental construction of landmark-based and topological model of indoor environments by a mobile robot. In: *IEEE Int. Conf. on Robotics and Automation*, IEEE Computer Society Press, Los Alamitos (1996)
6. Choset, H., Lynch, K.M., Hutchinson, S., Cantor, G., Burgard, W., Kavraky, L.E., Thrun, S.: *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge (2005)
7. Dubins, L.E.: On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal position and tangents. *American Journal of Mathematics* 79, 497–516 (1957)
8. Hayet, J.B., Lerasle, F., Devy, M.: A Visual Landmark Framework for Mobile Robot Navigation. *Image and Vision Computing* 25(8), 1341–1351 (2007)

9. Laumond, J.P.: Feasible trajectories for mobile robots with kinematic and environment constraints. In: Hertzberger, L.O., Groen, F.C.A. (eds.) *Intelligent Autonomous Systems*, pp. 346–354. North-Holland, New York (1987)
10. Latombe, J.C.: *Robot motion planning*. Kluwer academic publishers, Dordrecht (1991)
11. Laumond, J.P., Jacobs, P.E., Taix, M., Murray, R.M.: A Motion planner for non-holonomic mobile robots. *IEEE Trans. on Robotics and Automation* 10(5), 577–593 (1994)
12. Laumond, J.P.: *Robot Motion Planning and Control*. Springer, Toulouse France (1998)
13. LaValle, S.M.: *Planning Algorithms*. Cambridge University Press, Cambridge (2006)
14. Lazanas, A., Latombe, J.-C.: Landmark-based robot navigation. *Algorithmica* 13, 472–501 (1995)
15. Li, Z., Canny, J.: Motion of two rigid bodies with rolling constraint. *IEEE Trans. on Robotics and Automation* 6, 62–72 (1990)
16. Murray, R.M., Sastry, S.S.: Nonholonomic motion planning: Steering using sinusoids. *IEEE Trans. on Robotics and Automation* 38(5), 700–716 (1993)
17. Murrieta-Cid, R., Parra, C., Devy, M.: Visual Navigation in Natural Environments: From Range and Color Data to a Landmark-based Model. *Journal Autonomous Robots* 13(2), 143–168 (2002)
18. Soueres, P., Laumond, J.P.: Shortest path synthesis for a car-like robot. *IEEE Trans. on Autom Control* 14(5), 672–688 (1996)
19. Reeds, J.A., Shepp, R.A.: Optimal Paths for a car that goes both forward and backwards. *Pacific Journal of Mathematics* 145(2), 367–393 (1990)
20. Sussmann, H.J., Liu, W.: Limits of highly oscillatory controls and the approximation of general paths by admissible trajectories, Tech. Rep. SYSCON-91-02., Rutgers Center for Systems and Control (February 1991)
21. Thrun, S.: Bayesian landmark learning for mobile robot localization. *Machine Learning* 33(1), 41–76 (1998)

# Learning Performance in Evolutionary Behavior Based Mobile Robot Navigation

Tomás Arredondo V., Wolfgang Freund, César Muñoz, and Fernando Quirós

Universidad Técnica Federico Santa María, Valparaíso, Chile,  
Departamento de Electrónica,  
Casilla 110 V, Valparaíso, Chile  
tarredondo@elo.utfsm.cl

**Abstract.** In this paper we utilize information theory to study the impact in learning performance of various motivation and environmental configurations. This study is done within the context of an evolutionary fuzzy motivation based approach used for acquiring behaviors in mobile robot exploration of complex environments. Our robot makes use of a neural network to evaluate measurements from its sensors in order to establish its next behavior. Adaptive learning, fuzzy based fitness and Action-based Environment Modeling (AEM) are integrated and applied toward training the robot. Using information theory we determine the conditions that lead the robot toward highly fit behaviors. The research performed also shows that information theory is a useful tool in analyzing robotic training methods.

## 1 Introduction

Navigation in unknown and unstructured environments is of fundamental interest in mobile robotics. Numerous and extensive investigations have been made into this problem with the aid of various robotic architectures, sensors and processors. Behavior based architectures (e.g. subsumption) is an approach that in general does not support the use of world models or symbolic knowledge. This design philosophy promotes the idea that robots should be inexpensive, robust to sensor and other noise, incremental, uncalibrated and without complex computers and communication systems. Behavior based learning systems typically include reinforcement learning, neural networks, genetic algorithms, fuzzy systems, case and memory based learning [1]. As is the case with natural organisms, these biologically based mechanisms are capable of avoiding local minima, have the ability to extrapolate and are resilient to noise.

Recent research in behavior based robotics has focused on providing more natural and intuitive interfaces between robots and people [2,3]. A motivation based approach, introduced in [4] follows this trend by decoupling specific robot behavior using an intuitive interface based on biological motivations (e.g. curiosity, hunger, etc) [5].

It is a well known fact in machine learning that having diversity during training can provide for the emergence of more robust and adaptive systems capable of coping with a variety of environmental challenges [6,7]. To the best of our knowledge a quantitative method of measuring diversity in this context is not currently available. Toward this goal we propose using entropy based methods for measuring motivation and environmental diversity. Using these entropy based measures we investigate the effects of environmental and motivation diversity on robotic fitness.

In our experiments robot navigation is performed in a simulator [8] by providing sensor values directly into a neural network that drives left and right motors. The robot uses infrared sensors which give limited information about the surroundings in which the robot is located. Action-based environmental modeling (AEM) is implemented with a small action set of four basic actions (e.g. go straight, turn left, turn right, turn around) in order to encode a sequence of actions based on sensor values. The search space of behaviors is huge and designing suitable behaviors by hand is very difficult [9] therefore a genetic algorithm (GA) is used within the simulator to find appropriate behaviors. The GA selects from a population of robots (neural networks) using a fuzzy fitness function that considers various robotic motivations such as: the need for exploration (curiosity), the need to conserve its battery (energy), the desire to determine its location (orientation), and the capacity to return to its initial position (homing).

This paper is organized as follows. Section 2 gives a description of the robotic system and the entropy measures used for diversity evaluation. Section 3 introduces the experiments performed. In sections 4 and 5 we describe and summarize our results. Finally, in section 6 some conclusions are drawn.

## 2 Entropy Measures and Robotics System Description

This section presents the entropy based measures used for diversity measurements as well as the robotic system used for these studies. The system has several different elements including: the robot simulator, action based environmental modeling, neural networks, GA, and fuzzy logic based fitness.

### 2.1 Robot Configuration

For our experiments, a small simple robot was used. The robot configuration has two DC motors and eight (six front and two back) infrared proximity sensors used to detect nearby obstacles. These sensors provide 10 bit output values (with 6% random noise), which allow the robot to know in approximate form the distance to local obstacles. The simulator provides the readings for the robot sensors according to the robot position and the map (room) it is in. The simulator also has information for the different areas that the robot visits and the various obstacles detected in the room.



In our implementation, the robot generates a zone map in which the zones are marked with various values: obstacles are indicated with a value of -1, those not visited by the robot are marked with a 0 and the visited ones with a 1. In order to navigate, the robot executes 500 steps in each experiment, but not every step produces forward motion as some only rotate the robot. The robot is not constrained by its battery since 100% charge level allows more than 1000 steps.

## 2.2 Action-Based Environmental Modeling

To reduce the search space of behaviors, we use a limited number of actions for the robot to execute in each step. Using AEM based encoding four basic actions are used:

- A1: Go 55 mm straight on.
- A2: Turn 30°left.
- A3: Turn 30°right.
- A4: Turn 180°right.

In AEM, a SOM [10] network is used by the robot to determine the room he is navigating in (localization). Robot navigation produces actions which are saved as action sequences. These action sequences are converted using chain coding into an environment vector. These vectors are fed into the SOM network for unsupervised learning. After training the SOM network associates a vector with one of its output nodes (*r*-nodes) [9]. We used inputs of 1000 steps for all rooms used in training, these were alternately presented to the SOM network for 10000 iterations, the network had a linear output layer of 128 *r*-nodes indicating the maximum possible number of rooms that could be identified.

## 2.3 Artificial Neural Network

The neural network used in the robot is of the feed forward type with eight input neurons (one per sensor), five neurons in the hidden layer and two output neurons directly connected to the motors that produce the robot movement.

## 2.4 Genetic Algorithm

A GA is used to find an optimal configuration of weights for the neural network. Each individual in the GA represents a neural network which is evolving with the passing of different generations. The GA uses the following parameters:

- Population size: 200.
- Crossover operator: Random crossover.
- Selection method: Elite strategy selection.
- Mutation rate  $P_{mut}$ : 1%.
- Generations: 90.

## 2.5 Fuzzy Fitness Calculation

We have used fuzzy logic (Fig. 11) toward implementing a motivation based interface for determining robotic fitness. The motivation set ( $M$ ) considered in this study includes: homing ( $m_1$ ), curiosity ( $m_2$ ), energy ( $m_3$ ), and orientation ( $m_4$ ). These motivations are used as input settings (between 0 and 1) prior to running each experiment.

In calculating robotic fitness, four fuzzy variables with five membership functions each are used ( $4^5 = 1024$  different fuzzy rules). The set of fitness criteria and the fuzzy variables that correspond to them are: proper action termination and escape from original neighborhood area ( $f_1$ ), amount of area explored ( $f_2$ ), percent of battery usage ( $f_3$ ) and environment recognition ( $f_4$ ). The values for these criteria are normalized (range from 0 to 1) and are calculated after the robot completes each run:

- $f_1$ : a normalized final distance to home
- $f_2$ : percentage area explored relative to the optimum
- $f_3$ : estimated percent total energy consumption considering all steps taken
- $f_4$ : determined by having the robot establish which room he is in ( $r$ -node versus the correct one), using the previously trained SOM network.

The membership functions used are given in Fig. 11. Sample fuzzy rules (numbers 9 and 10) are given as follows (here the  $K$  array is a simple increasing linear function):

if ( $f_1 == H$ ) and ( $f_2 == L$ ) and ( $f_3 == V.L.$ ) and ( $f_4 == V.L.$ ) then

$$f[9] = m_1 f_1 K[4] + m_2 f_2 K[2] + m_3 f_3 K[1] + m_4 f_4 K[1]$$

if ( $f_1 == V.H.$ ) and ( $f_2 == L$ ) and ( $f_3 == V.L.$ ) and ( $f_4 == V.L.$ ) then

$$f[10] = m_1 f_1 K[5] + m_2 f_2 K[2] + m_3 f_3 K[1] + m_4 f_4 K[1]$$

During training, a run environment (room) is selected and the  $GA$  initial robot population is randomly initialized. After this, each robot in the population performs its task (navigation and optionally environment recognition) and a set of fitness values corresponding to the performed task are obtained. Finally, robotic fitness is calculated using the fitness criteria information provided by the simulator and the different motivations at the time of exploration.

## 2.6 Entropy Measures

Entropy is used to measure motivation and environmental diversity. Our concept of diversity follows the well established definition of entropy as a measure of the uncertainty (which generates a diversity of outcomes) in a system [11].

Let us define a motivation set  $M$  as  $\{m_1, m_2, \dots, m_n\}$ . Toward the calculation of motivation diversity  $H(M)$ , we consider the corresponding probabilities for  $\{m_1, m_2, \dots, m_n\}$  as  $\{p_1, p_2, \dots, p_n\}$ . We compute the entropy of the random variable  $M$  using:

$$H(M) = - \sum_{i=1}^n p_i \log(p_i), \text{ where } \sum_{i=1}^n p_i = 1.$$

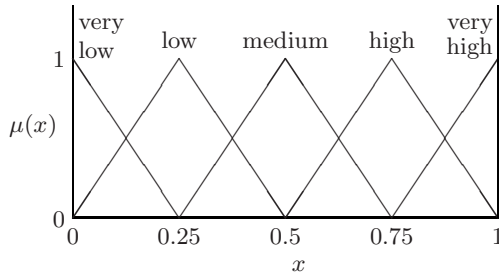


Fig. 1. Fuzzy membership functions

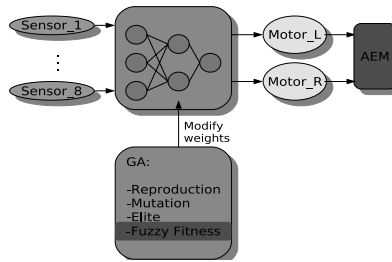


Fig. 2. System Overview

Note that when  $H(M) = 0$  the motivation is considered to have no diversity and as  $H(M) = 2$  it has maximal diversity.

Environmental entropy is calculated using an extension of the local image entropy method [12]. In this method an estimation of the information content of a pixel  $(x, y)$  based on a histogram of its neighborhood  $(w, h)$  pixel values is calculated as:

$$E_{w,h}(x, y) = - \sum_k p_{w,h}(k) \log p_{w,h}(k).$$

A part of an image (e.g. neighborhood or window) is interpreted as a signal of  $k$  different states with the local entropy  $E_{w,h}(x, y)$  determining the observers uncertainty about the signal [12].

Extending this method to a complete image (e.g. environment), and to obtain a measure of its diversity, we compute the average local image entropy (ALIE) for all pixels in the room. Following the definition of entropy, neighborhood width and height were set at twice the robots diameter so that the robot should have close to a maximum uncertainty of traversing through the neighborhood when  $E_{w,h}(x, y) = 0.5$ . Clearly if a neighborhood is empty or full the robot will have either no difficulty or no chance of traversing the terrain and hence the uncertainty for that neighborhood will be zero.

The obstacles used in the room were chosen at least as large as the robot size so as to preclude small checkered patterns which could also result in a

local entropy value near 0.5 but would not make any sense in terms of our definition of traversing uncertainty (because the robot clearly could not cross such a neighborhood). Also the obstacles were placed so that no obstacle would cause an access restriction to an area of the map.

### 3 Experimental Evaluation

Of primary interest was to study the impact of diversity on the robot's ability to learn behaviors. Two types of diversity were analyzed: environmental training diversity and motivation diversity.

All experiments had a training phase, in order to obtain the best weights for the NN, followed by a testing phase. The GA settings used in the training phase are given in section 2.4. The number of steps for each iteration was set to 1000 for the training phase, and 500 for the testing phase. The robot radius was 68.75 mm with a forward step size of 55 mm. The rooms are square (sides of 2750 mm) with various internal configurations of walls and obstacles. For mapping purposes rooms are divided into 2500 zones (each 55 mm by 55 mm).

#### 3.1 First Experiment: Environmental Diversity

For this experiment, we tested training in 15 different square rooms:  $r_1 - r_{15}$ . These rooms have an increasing number of obstacles in a fairly random distribution with the condition that any one obstacle should not preclude the robot from reaching any area of the room. Room  $r_1$  has no obstacles,  $r_2$  has only obstacle 1,  $r_3$  has obstacle 1 (in the same place as in  $r_2$ ) plus obstacle 2, and so on until  $r_{15}$  which contains obstacles 1 - 14. A room layout with the position of the obstacles (identified by its number) is shown in Fig. 3. Table 1, gives the computed average local entropy for all rooms.

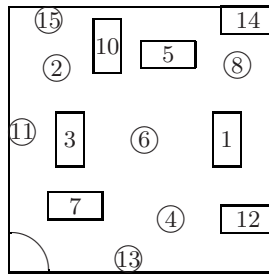


Fig. 3. Experiment 1 room layout ( $r_1-r_{15}$ )

In order to evaluate the impact of environmental training diversity over the robots navigation behavior, we trained the robot in each room for 90 generations. We set the fuzzy motivations ( $m_1, m_2, m_3, m_4$ ) as  $(0, 1, 0, 0)$ . After the

**Table 1.** Average local entropy for all rooms

Room	$H$	Room	$H$	Room	$H$
$r_1$	0	$r_6$	0.161	$r_{11}$	0.290
$r_2$	0.039	$r_7$	0.182	$r_{12}$	0.331
$r_3$	0.061	$r_8$	0.222	$r_{13}$	0.356
$r_4$	0.100	$r_9$	0.243	$r_{14}$	0.376
$r_5$	0.121	$r_{10}$	0.161	$r_{15}$	0.386

training phase, the best individual from the GA population was selected to run its respective testing phase in rooms  $r_1 - r_{10}$ . During testing, rooms  $r_{11} - r_{15}$  produced low fitness in all experiments, because of this small contribution we discarded these results from our analysis.

### 3.2 Second Experiment: Motivation Diversity

For this experiment, in order to see the effect of motivation diversity, we used four sets of fuzzy motivation criteria. Motivations ( $M$ ) ranged from highly focused to more diverse. In order to have a more objective comparison, indicated behavior score values are a weighted combination (according to  $M$ ) of the obtained behavior fitnesses  $\{f_1, f_2, f_3, f_4\}$ .

Training consisted of consecutively training the population in three different randomly selected rooms (based on our own office layouts)  $r_a - r_c$ . The population was trained for 30 generations in each of the three rooms in order. During testing the best individual of the 90 generations was tested in six rooms  $r_a - r_f$ . The various fitness values  $f_1 - f_4$  were calculated and used for fuzzy fitness calculation.

## 4 Experimental Results

Ten complete runs were performed of each experiment (each run consisting of one training and 10 test executions) and only average values are reported in our results.

### 4.1 First Experiment: Environmental Diversity

In Fig 4 we show the results of the testing phase after applying the respective training method specified for the environmental diversity experiment.

### 4.2 Second Experiment: Motivation Diversity

In tables 2- 5 we show the results of the testing phase after applying the respective training method specified for the motivation diversity experiment.

**Table 2.** Exp. 2: Behaviors for  $M$  as  $(0, 1, 0, 0)$ ,  $H(M) = 0$

Room	Score	Homing	% Exploration	% Battery usage
$r_a$	0.92	0.44	91.48	79.18
$r_b$	0.93	0.67	92.92	79.32
$r_c$	0.91	0.56	91.23	79.60
$r_d$	0.92	0.48	91.82	78.73
$r_e$	0.92	0.49	91.68	79.14
$r_f$	0.89	0.48	88.82	79.47
Avg.	0.91	0.52	91.33	79.24

**Table 3.** Exp. 2: Behaviors for  $M$  as  $(0.45, 0.55, 0, 0)$ ,  $H(M) = 0.99$

Room	Score	Homing	% Exploration	% Battery usage
$r_a$	0.86	0.91	82.50	80.13
$r_b$	0.89	0.93	85.03	81.49
$r_c$	0.89	0.95	84.28	81.31
$r_d$	0.86	0.92	81.50	81.46
$r_e$	0.90	0.93	86.75	81.44
$r_f$	0.87	0.94	80.94	81.30
Avg.	0.88	0.93	83.50	81.19

**Table 4.** Exp. 2: Summary of Behaviors for  $(M)$  as  $(.25, .5, .25, .0)$ ,  $H(M) = 1.50$

Room	Score	Homing	% Exploration	% Battery usage
$r_a$	0.83	0.83	84.85	79.49
$r_b$	0.88	0.96	88.14	80.88
$r_c$	0.88	0.94	87.65	81.41
$r_d$	0.85	0.86	85.92	80.67
$r_e$	0.85	0.83	88.08	81.14
$r_f$	0.86	0.91	84.94	81.42
Avg.	0.86	0.89	86.60	80.84

**Table 5.** Exp. 2: Behaviors for  $(M)$  as  $(0.09, 0.41, 0.09, 0.41)$ ,  $H(M) = 1.67$

Room	Score	Homing	% Exploration	% Battery usage	% Recognition
$r_a$	0.86	0.43	81.38	80.44	100
$r_b$	0.50	0.68	88.26	80.77	0
$r_c$	0.48	0.61	85.20	80.75	0
$r_d$	0.63	0.46	76.52	75.59	50
$r_e$	0.90	0.62	87.22	80.89	100
$r_f$	0.46	0.57	82.08	80.39	0
Avg.	0.64	0.56	83.44	79.81	41.67

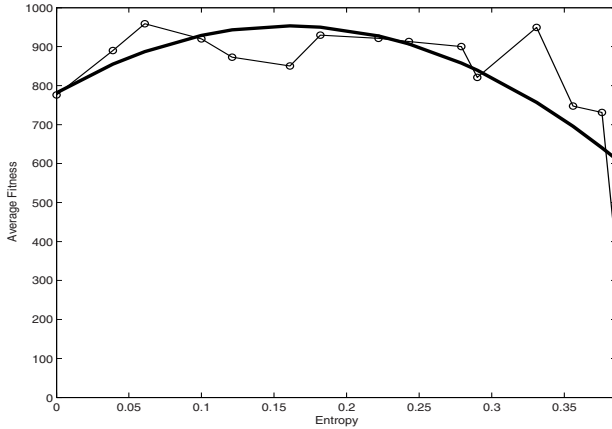


Fig. 4. Learning behavior given training rooms entropy

## 5 Discussion

Average local image entropy was shown as an effective measure of a training environments potential toward producing highly fit robots. As seen in Fig. 4 the average fitness obtained during testing is clearly dependent on training room average local image entropy. Interestingly, a training environment with too much environmental diversity is as unsuitable as one with not enough diversity. Our selection of neighborhood size was a reasonable one given the results obtained.

In general higher motivation diversity ( $H(M)$ ) caused lower average score values. This could be attributed to the many different (e.g. conflicting or non orthogonal) requirements of the different motivations upon a small robotic "brain". Even though obtained fitness was generally lower with more diverse motivations, the obtained behaviors demonstrated very good capability (e.g. room exploration, battery usage, etc) and were in close agreement with the specified motivations. In our experiments, environment recognition (e.g. SOM) was generally successful in determining the robots location. Motivation diversity results are somewhat counter intuitive in that simply by diversifying motivation values one could expect higher overall fitness but this is clearly not the case due to robotic system constraints.

## 6 Conclusions

Fuzzy motivations provide an intuitive and user friendly mechanism of obtaining a wide range of different behaviors. Our entropy based measures are a useful tool toward analyzing complex systems such as robotic training environments. The average local entropy measure shows promise as a signal information metric with potential usage in other fields.

Using these methods in various navigation experiments it was possible to see and contrast the effects of training and motivation diversity. Future work includes utilizing fuzzy motivations within hybrid architectures and parametric studies (e.g. linkages between motivations). We are currently working on implementation of these methods in physical robots.

## Acknowledgements

This research was partially funded by the DGIP of UTFSM (230726).

## References

1. Arkin, R.: Behavior-Based Robotics. MIT Press, Cambridge (1998)
2. Park, H., Kim, E., Kim, H.: Robot Competition Using Gesture Based Interface. In: Hromkovič, J., Nagl, M., Westfechtel, B. (eds.) WG 2004. LNCS, vol. 3353, pp. 131–133. Springer, Heidelberg (2004)
3. Jensen, B., Tomatis, N., Mayor, L., Drygałło, A., Siegwart, R.: Robots Meet Humans - Interacion in Public Spaces. IEEE Transactions on Industrial Electronics 52(6), 1530–1546 (2006)
4. Arredondo, T., Freund, W., Muñoz, C., Navarro, N., Quirós, F.: Fuzzy Motivations for Evolutionary Behavior Learning by a Mobile Robot. In: Ali, M., Dapoigny, R. (eds.) IEA/AIE 2006. LNCS (LNAI), vol. 4031, pp. 462–471. Springer, Heidelberg (2006)
5. Huitt, W.: Motivation to learn: An overview. Educational Psychology Interactive (2001), <http://chiron.valdosta.edu/whuitt/col/motivation/motivate.html>
6. Tan, K.C., Goh, C.K., Yang, Y.J., Lee, T.H.: Evolving better population distribution and exploration in evolutionary multi-objective optimization. European Journal of Operations Research 171, 463–495 (2006)
7. Chalmers, D.J.: The evolution of learning: An experiment in genetic connectionism. In: Proceedings of the 1990 Connectionist Models Summer School, pp. 81–90. M. Kaufmann, San Mateo, CA (1990)
8. YAKS simulator website: <http://www.his.se/iki/yaks>
9. Yamada, S.: Recognizing environments from action sequences using self-organizing maps. Applied Soft Computing 4, 35–47 (2004)
10. Teuvo, K.: The self-organizing map. Proceedings of the IEEE 79(9), 1464–1480 (1990)
11. Cover, T., Thomas, J.: Elements of Information Theory. Wiley, New York (1991)
12. Handmann, U., Kalinke, T., Tzomakas, C., Werner, M., Weelen, W.v.: An image processing system for driver assistance. Image and Vision Computing 18, 367–376 (2000)



# Fuzzifying Clustering Algorithms: The Case Study of MajorClust

Eugene Levner<sup>1</sup>, David Pinto<sup>2,3</sup>, Paolo Rosso<sup>2</sup>,  
David Alcaide<sup>4</sup>, and R.R.K. Sharma<sup>5</sup>

<sup>1</sup>Holon Institute of Technology, Holon, Israel

<sup>2</sup>Department of Information Systems and Computation, UPV, Spain

<sup>3</sup>Faculty of Computer Science, BUAP, Mexico

<sup>4</sup>Universidad de La Laguna, Tenerife, Spain

<sup>5</sup>Indian Institute of Technology, Kanpur, India

levner@hit.ac.il, {dpinto, proso}@dsic.upv.es,  
dalcaide@ull.es, rrrks@iitk.ac.in

**Abstract.** Among various document clustering algorithms that have been proposed so far, the most useful are those that automatically reveal the number of clusters and assign each target document to exactly one cluster. However, in many real situations, there not exists an exact boundary between different clusters. In this work, we introduce a fuzzy version of the MajorClust algorithm. The proposed clustering method assigns documents to more than one category by taking into account a membership function for both, edges and nodes of the corresponding underlying graph. Thus, the clustering problem is formulated in terms of weighted fuzzy graphs. The fuzzy approach permits to decrease some negative effects which appear in clustering of large-sized corpora with noisy data.

## 1 Introduction

Clustering analysis refers to the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait, often proximity, according to some defined distance measure [1,2,3]. Clustering methods are usually classified with respect to their underlying algorithmic approaches: hierarchical, iterative (or partitional) and density based are some instances belonging to this classification. Hierarchical algorithms find successive clusters using previously established ones, whereas partitional algorithms determine all clusters at once. Hierarchical algorithms can be agglomerative (“bottom-up”) or divisive (“top-down”); agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. Iterative algorithms start with some initial clusters (their number either being unknown in advance or given a priori) and intend to successively improve the existing cluster set by changing their “representatives” (“centers of gravity”, “centroids”) , like in K-Means [3] or by iterative node-exchanging (like in [4]).

An interesting density-based algorithm is MajorClust [5], which automatically reveals the number of clusters, unknown in advance, and successively increases the total “strength” or “connectivity” of the cluster set by cumulative attraction of nodes between the clusters.

MajorClust [5] is one of the most promising and successful algorithms for unsupervised document clustering. This graph theory based algorithm assigns each document to that cluster the majority of its neighbours belong to. The node neighbourhood is calculated by using a specific similarity measure which is assumed to be the weight of each edge (similarity) between the nodes (documents) of the graph (corpus). MajorClust automatically reveals the number of clusters and assigns each target document to exactly one cluster. However, in many real situations, there not exists an exact boundary between different categories. Therefore, a different approach is needed in order to determine how to assign a document to more than one category. The traditional MajorClust algorithm deals with crisp (hard) data, whereas the proposed version, called *F*-MajorClust, will use fuzzy data. We suggest to calculate a weighed fuzzy graph with edges between any pairs of nodes that are supplied with fuzzy weights which may be either fuzzy numbers or linguistic variables. Thereafter, a fuzzy membership function will be used in order to determine the possible cluster a node belongs to. The main feature of the new algorithm, *F*-MajorClust, which differs from MajorClust, is that all the items (for example, the documents to be grouped) are allowed to belong to two and more clusters.

The rest of this paper is structured as follows. The following section recalls the case study of the K-means algorithm and its fuzzy version. In Section [3] the traditional MajorClust algorithm is described whereas its fuzzy version is discussed in Section [4]. Finally we give some conclusions and further work.

## 2 *K*-Means and Fuzzy *K*-Means Clustering

The widely known *K*-means algorithm assigns each point to the cluster whose center is nearest. The center is the average of all the points of the cluster. That is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster. The algorithm steps are ([3]):

1. Choose the number  $k$  of clusters.
2. Randomly generate  $k$  clusters and determine the cluster centers, or directly generate  $k$  random points as cluster centers.
3. Assign each point to the nearest cluster center.
4. Recompute the new cluster centers.
5. Repeat the two previous steps until some convergence criterion is met (usually that the assignment has not changed).

The main advantages of this algorithm are its simplicity and speed which allows it to run on large datasets. This explains its wide applications in very many areas. However, a number of questions arise of which some examples follow. Since the choice of centers and assignment are random, and, hence, resulting clusters

can be different with each run, how to yield the best possible result? How can we choose the “best” distance  $d$  among very many possible options? How can we choose the “best” center if we have many possible competing variants? How do we choose the number  $K$  of clusters, especially in large-scale data bases? Having found multiple alternative clusterings for a given  $K$ , how can we then choose among them?

Moreover,  $K$ -means has several pathological properties and limitations. The algorithm takes account only of the distance between the centers and the data points; it has no representation of the weight or size of each cluster. Consequently,  $K$ -means behaves badly if several data bases strongly differ in size; indeed, data points that actually belong to the broad cluster have a tendency to be incorrectly assigned to the smaller cluster (see [1], for examples and details). Further, the  $K$ -means algorithm has no way of representing the size or shape of a cluster. In [1], there given an example when the data naturally fall into two narrow and elongated clusters. However, the only stable state of the  $K$ -means algorithm is that the two clusters will be erroneously sliced in half. One more criticism of  $K$ -means is that it is a ‘crisp’ rather than a ‘soft’ algorithm: points are assigned to exactly one cluster and all points assigned to a cluster are equals in that cluster. However, points located near the border between two or more clusters should, apparently, play a partial role in several bordering clusters. The latter disadvantage is overcome by Fuzzy  $K$ -means described below, whereas another mentioned-above disability of the  $K$ -means associated with the size and shape of clusters, is treated by the algorithm  $F$ -MajorClust presented in Section 4. In fuzzy clustering, data elements may belong to more than one cluster, and associated with each element we have a set of membership levels. These indicate a degree of belonging to clusters, or the “strength” of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters. Thus, border nodes of a cluster, may be in the cluster to a lesser degree than inner nodes. For each point  $x$  we have a coefficient giving the degree of being in the  $k$ -th cluster  $u_k(x)$ . Usually, the sum of those coefficients is defined to be 1:

$$\forall x \quad \sum_{k=1}^{NumClusters} u_k(x) = 1 \tag{1}$$

In the fuzzy  $C$ -Means (developed by Dunn in 1973 [6] and improved by Bezdek in 1981 [7]), the center (called “centroid”) of a cluster is the mean of all points, weighted by their degree of belonging to the cluster:

$$center_k = \frac{\sum_x u_k(x)^m x}{\sum_x u_k(x)^m} \tag{2}$$

where  $m$  is a fuzzification exponent; the larger the value of  $m$  the fuzzier the solution. At  $m = 1$ , fuzzy  $C$ -Means collapses to the crisp  $K$ -means, whereas at very large values of  $m$ , all the points will have equal membership with all the clusters. The degree of belonging is related to the inverse of the distance to the cluster:

$$u_k(x) = \frac{1}{d(\text{center}_k, x)}, \tag{3}$$

then the coefficients are normalised and fuzzyfied with a real parameter  $m > 1$  so that their sum is 1. Therefore,

$$u_k(x) = \frac{1}{\sum_j d(\text{center}_k, x)d(\text{center}_j, x)}, \tag{4}$$

for  $m$  equal to 2, this is equivalent to linearly normalise the coefficient to make the sum 1. When  $m$  is close to 1, then the closest cluster center to the point is given a greater weight than the others. The fuzzy  $C$ -Means algorithm is very similar to the  $K$ -means algorithm: Its steps are the following ones:

1. Choose a number of clusters.
2. Assign randomly to each point coefficients  $u_k(x)$  for being in the clusters.
3. Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is no more than  $\epsilon$ , the given "sensitivity threshold"):
  - (a) Compute the centroid for each cluster, using the formula above.
  - (b) For each point, compute its coefficients  $u_k(x)$  of being in the clusters, using the formula above.

The fuzzy algorithm has the same problems as  $K$ -means: the results depend on the initial choice of centers, assignments and weights, and it has no way of taking the shape or size of the clusters into account. The algorithm MajorClust presented below is intended to overcome the latter disadvantage.

### 3 The Majorclust Clustering Algorithm

The algorithm is designed to find the cluster set maximizing the total cluster connectivity  $A(C)$ , which is defined in [5] as follows:

$$A(C) = \sum_{k=1}^K |C_k| \lambda_k, \tag{5}$$

where  $C$  denotes the decomposition of the given graph  $G$  into clusters,  $C_1, C_2, \dots, C_k$  are clusters in the decomposition  $C$ ,  $\lambda_k$  designates the edge connectivity of cluster  $G(C_k)$ , that is, the minimum number of edges that must be removed to make graph  $G(C_k)$  disconnected.

MajorClust operationalises iterative propagation of nodes into clusters according to the "maximum attraction wins" principle [8]. The algorithm starts by assigning each point in the initial set its own cluster. Within the following relabelling steps, a point adopts the same cluster label as the "weighted majority of its neighbours". If several such clusters exist, one of them is chosen randomly. The algorithm terminates if no point changes its cluster membership.

#### The MajorClust algorithm

**Input:** object set  $D$ , similarity measure  $\varphi : D \times D \rightarrow [0; 1]$ , similarity threshold  $\tau$ .

**Output:** function  $\delta : D \rightarrow N$ , which assigns a cluster label to each point.

1.  $i := 0$ , ready := false
2. for all  $p$  from  $D$  do  $i := i + 1$ ,  $\delta(p) := i$  enddo
3. while ready = false do
  - (a) ready := true
  - (b) for all  $q$  from  $D$  do
    - i.  $\delta^* := i$  if  $\Sigma\{\varphi(p, q) | \varphi(p; q) \geq t \text{ and } \delta(p) = i\}$  is maximum.
    - ii. if  $\delta(q) \neq \delta^*$  then  $\delta(q) := \delta^*$ , ready := false
  - (c) enddo
4. enddo

*Remark.* The similarity threshold  $\tau$  is not a problem-specific parameter but a constant that serves for noise filtering purposes. Its typical value is 0.3.

The MajorClust is a relatively new clustering algorithm with respect to other methods. However, its characteristic of automatically discovering the target number of clusters make it even more and more attractive [9,10,11,12], and hence the motivation of fuzzifying it.

In the following section, we will describe in detail the proposed fuzzy approach for the traditional MajorClust clustering algorithm.

## 4 Fuzzification of MajorClust

### 4.1 Fuzzy Weights of Edges and Nodes in *F*-MajorClust

The measure of membership of any edge  $i$  in a cluster  $k$  is presented by a membership function  $\mu_{ik}$ , where  $0 \leq \mu_{ik} \leq 1$ , and  $\sum_k \mu_{ik} = 1$  for any  $i$ .

In order to understand the way it is employed, we will need the following definitions. A node  $j$  is called *inner* if all its neighbours belong to the same cluster as the node  $j$ . If an edge  $i$  connects nodes  $x$  and  $y$ , we will say that  $x$  and  $y$  are the *end* nodes of the edge  $i$ . A node  $j$  is called *boundary* if some of its neighbours belong to a cluster (or several clusters) other than the cluster containing the node  $j$  itself.

The main ideas behind the above concept of the fuzzy membership function  $\mu_{ik}$  is that the edges connecting the inner nodes in a cluster may have a larger “degree of belonging” to a cluster than the “peripheral” edges (which, in a sense, reflects a greater “strength of connectivity” between a pair of nodes). For instance, the edges (indexed  $i$ ) connecting the *inner nodes* in a cluster (indexed  $k$ ) are assigned  $\mu_{ik} = 1$  whereas the edges linking the *boundary nodes* in a cluster have  $\mu_{ik} < 1$ . The latter dependence reflects the fact that in the forthcoming algorithm the boundary nodes have more chances to leave a current cluster than the inner ones, therefore, the “strength of connectivity” of a corresponding edge in the current cluster is smaller. As a simple instance case, we define  $\mu_{ik} = \frac{a_{ik}}{b_i}$ , where  $a_{ik}$  is the number of those neighbours of the end nodes of  $i$  that belong to the same cluster  $k$  as the end nodes of  $i$ , and  $b_i$  is the number of all neighbours to the end nodes of  $i$ . In a more advanced case, we define  $\mu_{ik} = \frac{A_{ik}}{B_i}$ , where  $A_{ik}$  is

the sum of the weights of edges linking the end nodes of  $i$  with those neighbours of the end nodes of  $i$  that belong to the same cluster  $k$  as the end nodes of  $i$ , and  $B_i$  is the total sum of the weights of the edges adjacent to the edge  $i$ .

Furthermore, we introduce the measure of membership of any item (node)  $j$  in any cluster  $k$ , which is presented by the membership function  $\gamma_{jk}$ , where  $0 \leq \gamma_{jk} \leq 1$ , and  $\sum_k \gamma_{jk} = 1$  for any  $j$ . Notice that these weights are assigned to nodes, rather than to the edges: this specific feature being absent in all previous algorithms of MajorClust type. The value of the membership function  $\gamma_{jk}$  reflects the *semantic correspondence* of node  $j$  to cluster  $k$ , and is defined according to the “fitness” of node  $j$  to cluster  $k$  as defined in [13]. The idea behind this concept is to increase the role of the nodes having a larger fitness to their clusters. In the formula (6) and the text below,  $\gamma_{jk}$  is a function of a cluster  $C_k$  containing the node  $j$ :  $\gamma_{jk} = \gamma_{jk}(C_k)$  which may dynamically change in the algorithm suggested below as soon as  $C_k$  changes. The objective function in the clustering problem becomes more general than that in [5] so that the weights of nodes are being taken into account, as follows:

$$\text{Maximize } \Lambda(C) = \sum_{k=1}^K \left( \sum_{j=1}^{|C_k|} \mu_{i(j),k} \lambda_k + \sum_{j=1}^n \gamma_{jk}(C_k) \right), \tag{6}$$

where  $C$  denotes the decomposition of the given graph  $G$  into clusters,  $C_1, \dots, C_K$  are not-necessarily disjoint clusters in the decomposition  $C$ ,  $\Lambda(C)$  denotes the total weighted connectivity of  $G(C)$ ,  $\lambda_k$  designates, as in MajorClust, the edge connectivity of cluster  $G(C_k)$ , the weight  $\mu_{i(j),k}$  is the membership degree of arc  $i(j)$  containing node  $j$  in cluster  $k$ , and finally,  $\gamma_{jk}(C_k)$  is the fitness of node  $j$  to cluster  $k$ .  $\lambda_k$  is calculated according to [5], meaning the cardinality of the set of edges of minimum total weight  $\sum_i \mu_{ik}$  that must be removed in order to make the graph  $G(C_k)$  disconnected.

#### 4.2 Limitations of MajorClust and *if-then* Rules

The fuzzy weights of edges and nodes (that is, in informal terms, the fuzzy semantic correlations between the documents and the fuzzy fitness of documents to categories) can be presented not only in the form of fuzzy numbers defined between 0 and 1 reflecting a flexible (fuzzy) measure of fitness (which sometimes is called “responsibility”). Moreover they even may be linguistic variables (e.g. small, medium, large, etc). In the latter case, they are assigned the so-called “grades” introduced in [14][13]. The presence of fuzzy weights on edges and nodes permit us to avoid several limitations of the standard MajorClust. The most important among them are the following ones:

1. When MajorClust runs, it may include nodes with weak links, i.e., with a small number of neighbours which inevitably leads to the decrease of the objective function already achieved (see [5]).
2. MajorClust assigns each node to that cluster the majority of its neighbours belong to, and when doing this, the algorithm does not specify the case

when there are several “equivalent” clusters equally matching the node. The recommendation in [5] to make this assignment in an arbitrary manner, may lead to the loss of a neighbouring good solution.

3. MajorClust scans nodes of the original graph in an arbitrary order, which may lead to the loss of good neighbouring solutions.
4. MajorClust takes into account only one local minimum among many others (which may lead to the loss of much better solutions than the one selected). These limitations will be avoided in the fuzzy algorithm suggested, by the price of greater computational efforts (the running time) and a larger required memory.

In the following, we list a set of *if-then* rules which may be added to the proposed fuzzy version of the MajorClust. The order of node scan is defined by the following decision rules R1-R3.

**Rule R1:** *If there are several nodes having majority in certain clusters (or the maximum value of the corresponding objective function), then choose first the node having the maximal number of neighbours.*

**Rule R2:** *If there are several nodes having both the majority and the maximum number of neighbours in certain clusters, then choose first the node whose inclusion leads to the maximum increase of the objective function.*

**Rule R2:** *If there are several nodes having both the majority and the maximum number of neighbours in certain clusters, then assign the nodes to clusters using the facility layout model taking into account the semantic fitness of nodes to clusters and providing the maximum increase of the objective function (the mathematical model formulation and computational approaches can be found in [15,16]).*

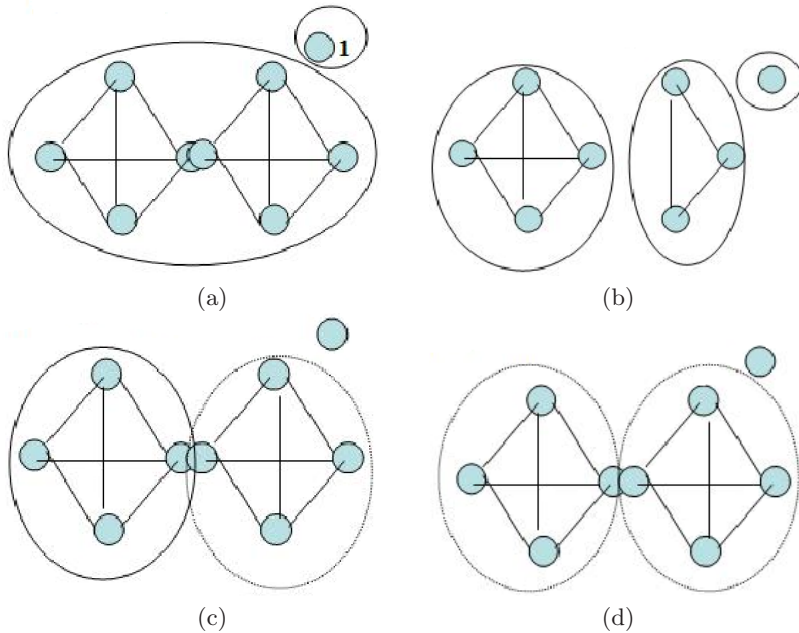
**Rule R3:** *If, at some iterative step, the inclusion of some node would lead to the decrease of the objective function, then this node should be skipped (that is, it will not be allocated into any new cluster at that step).*

The algorithm stops when the next iterative step does not change the clustering (this is Rule 4) or any further node move leads to deteriorating of the achieved quality (defined by the formula (6) (this is Rule 5) or according to other stopping rules R6-R7 (see below).

**Rule R6:** *If the number of steps exceeds the given threshold, then stop.*

**Rule R7:** *If the increase in the objective function at  $H$  current steps is less than  $\epsilon$  ( $H$  and  $\epsilon$  are given by the experts and decision makers in advance), then stop.*

The  $F$ -MajorClust algorithm starts by assigning each point in the initial set its own cluster. Within the following re-labelling steps, a point adopts the same cluster label as the “weighted majority of its neighbours”, that is, the node set providing the maximum value to the generalized objective function [13]. If several such clusters exist, say  $z$  clusters, then a point adopts all of them and attains the membership function  $\omega_{jk} = 1/z$  for all clusters. At each step, if point  $j$  belongs to  $y$  clusters, then  $\omega_{jk} = 1/y$ . The algorithm terminates if no point changes its cluster membership.



**Fig. 1.** Butterfly effect in fuzzy clustering. (a) and (b) use classical MajorClust, whereas (c) and (d) use the *F*-MajorClust approach (with different node membership functions).

### 4.3 The Butterfly Effect

In Fig. 1 we can observe the so-called “butterfly effect”, which appears when some documents (nodes) of the dataset (graph) may belong to more than one cluster, in this case the fuzzy algorithm works better than the crisp one. Fig. 1(a) and 1(b) depict an example when the classical MajorClust algorithm has found two clusters and, then, according to formula (5) this clustering of eight nodes obtains a score of 21 ( $C=7 \times 3 + 1 \times 0 = 21$ ). Figure 1(b) shows the case when the classical MajorClust algorithm has detected three clusters for the same input data providing a total score of 18 ( $C=4 \times 3 + 3 \times 2 + 1 \times 0 = 18$ ). On the other hand, Fig. 1(c) and 1(d) demonstrate how the fuzzy algorithm works when some nodes can belong simultaneously to several different clusters. We assume that the algorithm uses formula (6) where, for the simplicity, we take  $\gamma_{jk}(C_k) = 0$ ; even in this simplified case the fuzzy algorithm wins. Two variants are presented: in Figure 1(c) we consider the case when the membership values are shared equally between two cluster with the membership value 0.5; then the obtained score is 21 ( $C=2 \times ((3+0.5) \times 3) + 1 \times 0 = 21$ ). Note that the value of the objective function is here the same as in the case 1(a). However, if the documents (nodes) are highly relevant to the both databases with the membership function values 1 then the fuzzy algorithm yields a better score which is presented in Fig. 1(d):  $C=2 \times ((3+1) \times 3) + 1 \times 0=24$ . It worth noticing that this effect becomes even stronger if  $\gamma_{jk}(C_k) > 0$  in (6).



## 5 Conclusions and Further Work

Our efforts were devoted to employ fuzzy clustering on text corpora since we have observed a good performance of the MajorClust in this context. We have proposed a fuzzy version of the classical MajorClust clustering algorithm in order to permit overlapping between the obtained clusters. This approach will provide a more flexible use of the mentioned clustering algorithm. We consider that there exist different areas of application for this new clustering algorithm which include not only data analysis but also pattern recognition, spatial databases, production management, etc. in cases when any object can be assigned to more than a unique category.

Special attention in the definition and operation of the so called fuzzifier will be needed, since it controls the amount of overlapping among the obtained clusters and, it is well known that for those corpora with varying data density, noisy data and big number of target clusters, some negative effects may appear [17]. These effects can be mitigated by proper calibration of the membership functions in [13] with the help of fuzzy *if-then* rules and the standard Mamdani-type inference scheme (see [18,19]).

As future work, we plan to compare *F*-MajorClust with *C*-Means performance by using the Reuters collection. Moreover, as a basic analysis, it will be also interesting to execute the classical MajorClust over the same dataset.

## Acknowledgements

This work has been partially supported by the MCyT TIN2006-15265-C06-04 research project, as well as by the grants BUAP-701 PROMEP/103.5/05/1536, FCC-BUAP, DPI2001-2715-C02-02, MTM2004-07550, MTM2006-10170, and SAB2005-0161.

## References

1. MacKay, D.J.C.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
2. Mirkin, B.G.: Mathematical Classification and Clustering. Springer, Heidelberg (1996)
3. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proc. of 5-th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press, Berkeley (1967)
4. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. Bell Systems Technical Journal 49(2), 291–308 (1970)
5. Stein, B., Nigemman, O.: On the nature of structure and its identification. In: Widmayer, P., Neyer, G., Eidenbenz, S. (eds.) WG 1999. LNCS, vol. 1665, pp. 122–134. Springer, Heidelberg (1999)
6. Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. Journal of Cybernetics 3, 32–57 (1973)

7. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
8. Stein, B., Busch, M.: Density-based cluster algorithms in low-dimensional and high-dimensional applications. In: Proc. of Second International Workshop on Text-Based Information Retrieval, TIR 2005, pp. 45–56 (2005)
9. Stein, B., Meyer, S.: Automatic document categorization. In: Günter, A., Kruse, R., Neumann, B. (eds.) KI 2003. LNCS (LNAI), vol. 2821, pp. 254–266. Springer, Heidelberg (2003)
10. Alexandrov, M., Gelbukh, A., Rosso, P.: An approach to clustering abstracts. In: Montoyo, A., Muñoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 275–285. Springer, Heidelberg (2005)
11. Pinto, D., Rosso, P.: On the relative hardness of clustering corpora. In: Proc. of TSD 2007 Conference. LNCS, Springer, Heidelberg (to appear, 2007)
12. Neville, J., Adler, M., Jensen, D.: Clustering relational data using attribute and link information. In: Proc. of the Text Mining and Link Analysis Workshop, IJCAI 2003 (2003)
13. Levner, E., Alcaide, D., Sicilia, J.: Text classification using the fuzzy borda method and semantic grades. In: Proc. of WILF-2007 (CLIP-2007). LNCS (LNAI), vol. 4578, pp. 422–429. Springer, Heidelberg (2007)
14. Levner, E., Alcaide, D.: Environmental risk ranking: Theory and applications for emergency planning. Scientific Israel - Technological Advantages 8(1-2), 11–21 (2006)
15. Koopmans, T.C., Beckman, M.: Assignment problems and the location of economic activities. *Econometrica* 25, 53–76 (1957)
16. Singh, S.P., Sharma, R.R.K.: A review of different approaches to the facility layout problem. *International Journal of Advanced Manufacutring Technology* 30(5-6), 426–433 (2006), <http://dx.doi.org/10.1007/s00170-005-0087-9>
17. Klawonn, F., Höpnnner, F.: What is fuzzy about fuzzy clustering-understanding and improving the concept of the fuzzifier. *Advances in Intelligent Data Analysis V*, 254–264 (2003)
18. Mamdani, E.H.: Application of fuzzy logic to approximate reasoning using linguistic synthesis. In: Proc. of the sixth international symposium on Multiple-valued logic, pp. 196–202 (1976)
19. Zimmermann, H.J.: Fuzzy Sets, Decision Making and Expert Systems. Kluwer Academic Publishers, Boston (1987)

# Taking Advantage of the Web for Text Classification with Imbalanced Classes\*

Rafael Guzmán-Cabrera<sup>1,2</sup>, Manuel Montes-y-Gómez<sup>3</sup>,  
Paolo Rosso<sup>2</sup>, and Luis Villaseñor-Pineda<sup>3</sup>

<sup>1</sup>FIMEE, Universidad de Guanajuato, Mexico  
guzmanc@salamanca.ugto.mx

<sup>2</sup>DSIC, Universidad Politécnica de Valencia, Spain  
proso@dsic.upv.es

<sup>3</sup>LTL, Instituto Nacional de Astrofísica, Óptica y Electrónica, Mexico  
{mmontesg, villasen}@inaoep.mx

**Abstract.** A problem of supervised approaches for text classification is that they commonly require high-quality training data to construct an accurate classifier. Unfortunately, in many real-world applications the training sets are extremely small and present imbalanced class distributions. In order to confront these problems, this paper proposes a novel approach for text classification that combines under-sampling with a semi-supervised learning method. In particular, the proposed semi-supervised method is specially suited to work with very few training examples and considers the automatic extraction of untagged data from the Web. Experimental results on a subset of Reuters-21578 text collection indicate that the proposed approach can be a practical solution for dealing with the class-imbalance problem, since it allows achieving very good results using very small training sets.

## 1 Introduction

Nowadays there is a lot of digital information available from the Web. This situation has produced a growing need for tools that help people to find, filter and analyze all these resources. In particular, text classification [10], the assignment of free text documents to one or more predefined categories based on their content, has emerged as a very important component in many information management tasks.

The state-of-the-art approach for automatic text classification considers the application of a number of statistical and machine learning techniques, including regression models, Bayesian classifiers, support vector machines, nearest neighbor classifiers, neuronal networks and statistical methods driven by a hierarchical topic dictionary to mention some [1, 10, 3]. A major difficulty with this kind of supervised techniques is that they commonly require high-quality training data to construct an accurate classifier. Unfortunately, in many real-world applications the training sets are *extremely small* and even worse, they present *imbalanced class distributions*

---

\* This work was done under partial support of CONACYT-Mexico (43990) MCyT-Spain (TIN2006-15265-C06-04) and PROMEP (UGTO-121).

(i.e., the number of examples in some classes are significantly greater than the number of examples in the others).

In order to overcome these problems, recently many researches have been working on semi-supervised learning algorithms as well as on different solutions to the class-imbalance problem (for an overview refer to [2, 11]). On the one hand, it has been showed that by augmenting the training set with additional –unlabeled– information it is possible to improve the classification accuracy using different learning algorithms such as naïve Bayes [9], support vector machines [7], and nearest-neighbor algorithms [13]. On the other hand, it has also been demonstrated that by adjusting the number of examples in the majority or minority classes it is possible to tackle the suboptimal classification performance caused by the class-imbalance [6]. In particular, there is evidence that under-sampling, a method in which examples of the majority classes are removed, leads to better results than over-sampling, a method in which examples from the minority classes are duplicated [5].

In this paper we propose a novel approach for text classification with imbalanced classes that combines under-sampling and semi-supervised methods. The idea is to use under-sampling to balance an original imbalanced training set, and then apply a semi-supervised classification method to compensate the missing of information by adding new –highly discriminative– training instances.

The most relevant component of the proposed approach is the semi-supervised classification method. It mainly differs from previous methods in three main concerns. First, it is specially suited to work with *very* few training examples. Whereas previous methods consider hundreds of training examples, our method allows working with just groups of ten labeled examples per class. Second, it does not require a predefined set of unlabeled examples. It considers the automatic extraction of related untagged data from the Web. Finally, given that it deals with few training examples, it does not aim including a lot of additional information in the training phase; instead, it only incorporates a small group of examples that considerably augment the dissimilarities among classes.

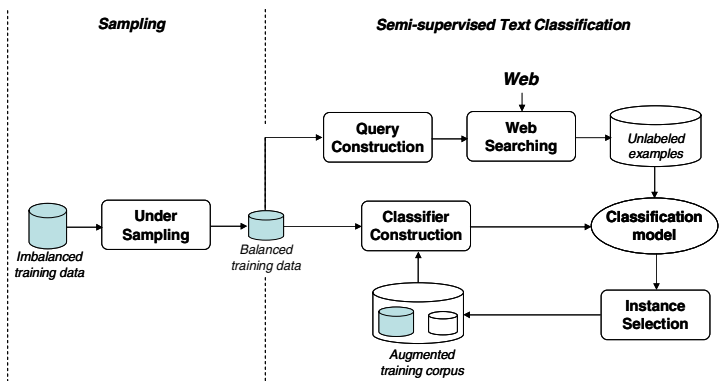


Fig. 1. General overview of the approach

It is important to mention that this method achieved very good results on classifying news documents about natural disasters [4]. It could construct an accurate classifier starting from only ten training examples per class. However, in that case, the training collection was simple: it only contained five clearly separable classes with no imbalance. In contrast, in this new experiment we aim to explore the capacity of the method to deal with more complex document collections that contain a great number of imbalanced and overlapped classes.

The rest of the paper is organized as follows. Section 2 shows the general scheme of the proposed approach for text classification with imbalanced classes. Section 3 describes the Web-based semi-supervised classification method. Section 4 presents some evaluation results on a subset of Reuters-21578 text collection. Finally, section 5 depicts our conclusions and future work.

## 2 Overview of the Proposed Approach

Figure 1 shows the general scheme of the proposed approach. It consists of two main phases: under-sampling and semi-supervised text classification.

Under-sampling is one of the methods most commonly used to adapt machine-learning algorithms to imbalanced classes. As we mentioned, it considers the elimination of training examples from the majority classes. In this case, examples to be removed can be randomly selected, or near miss examples, or examples that are far from the minority of the class instances [5].

In our particular case, we apply a kind of “extreme” under-sampling over the original data set. The idea is to assemble a *small* balanced training corpus by eliminating – at random – a great number of examples from all classes. This extreme strategy was mainly motivated by the fact that small training sets are more advantageous for our semi-supervised classification method. In addition, this decision was also motivated by our interest on demonstrating that the problem of learning from imbalanced classes can be modeled as one of learning from *very* small training sets.

The second phase considers the semi-supervised classification method. This method consists of two main processes. The first one deals with the corpora acquisition from the Web, while the second one focuses on the semi-supervised learning problem. The following section describes in detail these two processes.

It is important to point out that the Web has been lately used as a corpus in many natural language tasks [8]. In particular, Zelikovitz and Kogan [14] proposed a method for mining the Web to improve text classification by creating a background text set. Our proposal is similar to this approach in the sense of it also mines the Web for additional information (extra-unlabeled examples). Nevertheless, our method applies finer procedures to construct the set of queries related to each class and to combine the downloaded information.

## 3 Semi-supervised Text Classification

### 3.1 Corpora Acquisition

This process considers the automatic extraction of unlabeled examples from the Web. In order to do this, it first constructs a number of queries by combining the most

significant words for each class; then, using these queries it looks at the Web for some additional training examples related to the given classes.

**Query Construction.** In order to form queries for searching the Web, it is necessary to previously determine the set of relevant words for each class in the training corpus. The criterion used for this purpose is based on a combination of the frequency of occurrence and the information gain of words. We consider that a word  $w_i$  is relevant for class  $C$  if:

1. The frequency of occurrence of  $w_i$  in  $C$  is greater than the average occurrence of all words (happening more than once) in that class. That is:

$$f_{w_i}^C > \frac{1}{|C|} \sum_{\forall w \in C'} f_w^C, \text{ where } C' = \{w \in C \mid f_w^C > 1\}$$

2. The information gain of  $w_i$  with respect to  $C$  is positive. That is, if  $IG_{w_i}^C > 0$ .

Once obtained the set of relevant words per class, it is possible to construct the corresponding set of queries. Founded on the method by Zelikovitz and Kogan [14], we decide to construct queries of three words. This way, we create as many queries per class as all three-word combinations of its relevant words. We measure the significance of a query  $q = \{w_1, w_2, w_3\}$  to the class  $C$  as indicated below:

$$\Gamma_C(q) = \sum_{i=1}^3 f_{w_i}^C \times IG_{w_i}^C$$

**Web Searching.** The next action is using the defined queries to extract from the Web a set of additional unlabeled text examples. Based on the observation that most significant queries tend to retrieve the most relevant web pages, our method for searching the Web determines the number of downloaded examples per query in a direct proportion to its  $\Gamma$ -value. Therefore, given a set of  $M$  queries  $\{q_1, \dots, q_M\}$  for class  $C$ , and considering that we want to download a total of  $N$  additional examples per class, the number of examples to be extracted by a query  $q_i$  is determined as follows:

$$\Psi_C(q_i) = \frac{N}{\sum_{k=1}^M \Gamma_C(q_k)} \times \Gamma_C(q_i)$$

### 3.2 Semi-supervised Learning

As we previously mentioned, the purpose of this process is to increase the classification accuracy by gradually augmenting the originally small training set with the examples downloaded from the Web. Our algorithm for semi-supervised learning is an adaptation of a method proposed elsewhere [12]. It mainly considers the following steps:

1. Build a weak classifier ( $C_l$ ) using a specified learning method ( $l$ ) and the training set available ( $T$ ).
2. Classify the downloaded examples ( $E$ ) using the constructed classifier ( $C_l$ ). In order words, estimate the class for all downloaded examples.

3. Select the best  $m$  examples ( $E_m \subseteq E$ ) based on the following two conditions:
  - a. The estimate class of the example corresponds to the class of the query used to download it. In some way, this filter works as an ensemble of two classifiers:  $C_l$  and the Web (expressed by the set of queries).
  - b. The example has one of the  $m$ -highest confidence predictions.
4. Combine the selected examples with the original training set ( $T \leftarrow T \cup E_m$ ) in order to form a new training set. At the same time, eliminate these examples from the set of downloaded instances ( $E \leftarrow E - E_m$ ).
5. Iterate  $\sigma$  times over steps 1 to 4 or repeat until  $E_m = \emptyset$ . In this case  $\sigma$  is a user specified threshold.
6. Construct the final classifier using the enriched training set.

## 4 Experimental Evaluation

### 4.1 Experimental Setup

**Corpus.** We selected the subset of the 10 largest categories of the Reuters-21578 corpus. In particular, we considered the ModApte split distribution, which includes all labeled documents published before 04/07/87 as training data (i.e., 7206 documents) and all labelled documents published after 04/07/87 as testing set (i.e., 3220 documents). Table 1 shows some numbers on this collection.

**Table 1.** Training/testing data sets

Category	Training Set	Test Set
ACQ	1650	798
CORN	182	71
CRUDE	391	243
EARN	2877	1110
GRAIN	434	194
INTEREST	354	159
MONEY-FX	539	262
SHIP	198	107
TRADE	369	182
WHEAT	212	94
<i>Total</i>	<i>7206</i>	<i>3220</i>

**Searching the Web.** We used Google as search engine. We downloaded 2,400 additional examples (snippets for these experiments) per class.

**Learning method.** We selected naïve Bayes (NB) as the base classification method.

**Document Preprocessing.** We removed all punctuation marks and numerical symbols, that is, we only considered alphabetic tokens. We also removed stop words and hapax legomena, and converted all tokens to lowercase. On the other hand, in all experiments we took the 1000 most frequent words as classification features.

**Evaluation measure.** The effectiveness of the method was measured by the classification accuracy, which indicates the percentage of documents that have been correctly classified from the entire document set.

**Baseline.** For this case, all training data was used to construct a naïve Bayes classifier. The achieved accuracy of this classifier over the given test data was of 84.7%.

## 4.2 Experimental Results

As we mentioned in section 2, the proposed approach has two main phases: under-sampling and semi-supervised text classification. The idea is to apply under-sampling to assemble a balanced training corpus, and then use a semi-supervised classification method to compensate the missing of information by adding new –highly discriminative– training instances (i.e., snippets downloaded from the web).

Because our semi-supervised method is specially suited to work with *very few* training examples, we applied an “extreme” under-sampling over the original training data. Table 2 shows the accuracy results corresponding to different levels of data reduction. It is important to notice that using only 100 training examples per class it was practically possible to reach the baseline result.

**Table 2.** Accuracy percentage for different training data sets

Training examples per class	Accuracy percentage
10	58.6
20	73.7
30	77.3
40	79.3
50	81.8
80	82.8
100	84.1
182	84.0
<i>Baseline</i>	<i>84.7</i>

In order to evaluate the semi-supervised classification method we performed *two experiments*. The first one only used 10 training examples per class, whereas the other one employed 100 training instances per class.

It is important to clarify that using more examples allows constructing more general and consequently more relevant queries. For instance, using one hundred examples about the INTEREST class, we constructed queries such as: *<bank + money + interest>*, *<money + market + banks>* and *<bank + interest + rate>*.

Using the automatically constructed queries, we collected from the Web a set of 2,400 snippets per class, obtaining a total of 24,000 additional unlabeled examples. It is interesting to point out that thanks to the snippet’s small size (that only considers the immediate context of the query’s words), the additional examples tend to be less ambiguous and contain several valuable words that are highly related with the topic at hand. As an example, look at the following snippet for the class *INTEREST*:



*<compare mortgage rates home loans cd rates auto loans credit free  
objective information rate quotes consumer bank products cds auto  
loans home equity loans money market funds personal loans>*

Finally, the downloaded snippets were classified using the original document collection as training set (refer to section 3.2). The best *ten* examples per class, i.e., those with more confidence predictions, were selected at each iteration and were incorporated to the original training set in order to form a new training collection. In both experiments, we performed 10 iterations. Table 3 shows the accuracy results for all iterations of both experiments.

**Table 3.** Accuracy percentage after the training corpus enrichment

Labeled Training Instances	Base Accuracy	Iteration									
		1	2	3	4	5	6	7	8	9	10
10	58.6	66.9	68.7	69.6	70.3	<b>70.6</b>	68.6	69.0	69.0	68.5	68.7
100	84.1	84.6	84.7	84.8	86.6	86.8	86.8	<b>86.9</b>	86.7	86.7	86.7

From table 3 we can observe that the semi-supervised learning method did its job. For instance, when using only 10 training examples per class the method produced a notable 12% increase in the accuracy (from 56.6 to 70.6). Nevertheless, it is clear that given the complexity of the given test collection (that contains some semantically related classes such as grain, corn and wheat) it is necessary to start with more training examples.

In the case of the second experiment (which made use of 100 training examples per class), the increment in the accuracy was not as high as in the first experiment. It only moved the accuracy from 84% to 86.9%. However, it is important to mention that this increment was enough to outperform the baseline result. In other words, the method allowed obtaining a better accuracy using only 1000 training examples than considering all 7206.

## 5 Conclusions and Future Work

This paper proposed a novel approach for text classification with imbalanced classes that combines under-sampling and semi-supervised learning methods. The general idea of the approach is to use under-sampling to balance an original imbalanced training set, and then apply a semi-supervised classification method to compensate the missing of information by adding new –highly discriminative– training instances.

In particular, the most relevant component of the approach is the semi-supervised text classification method. This method differs from others in that: (i) it is specially suited to work with *very* few training examples, (ii) it automatically collects from the Web the unlabeled data and, (iii) it only incorporates into the training phase a small group of highly discriminative unlabeled examples.

In general, the achieved results allow us to formulate the following conclusions. On the one hand, the proposed combined approach can be a practical solution for the problem of text classification with imbalanced classes. On the other hand, our

Web-based semi-supervised learning method is a quite pertinent tool for text classification, since it allows achieving very good results using very small training sets.

As future work we plan to apply the proposed approach to other collections with higher imbalance rates, for instance, to a different subset of the Reuters corpus. Also, given that the highest accuracies were obtained before completing all possible iterations, we aim to study the behavior of the iterative semi-supervised learning process in order to define a better stop criterion. Finally, we also plan to evaluate the approach, in particular, the semi-supervised learning method, in some non-topical classification problems such as authorship attribution and genre detection.

## References

1. Aas, K., Eikvil, L.: Text Categorization: A survey, Technical Report, number 941, Norwegian Computing Center (1999)
2. Chawla, N.V., Japkowicz, N., Kolcz, A.: Editorial: Special Issue on Learning from Imbalanced data Sets. *ACM SIGKDD Exploration Newsletters* 6(1) (June 2004)
3. Gelbukh, A., Sidorov, G., Guzman-Arénas, A.: Use of a Weighted Topic Hierarchy for Document Classification. In: Matoušek, V., Mautner, P., Ocelíková, J., Sojka, P. (eds.) *TSD 1999*. LNCS (LNAI), vol. 1692, pp. 130–135. Springer, Heidelberg (1999)
4. Guzmán-Cabrera, R., Montes-y-Gómez, M., Rosso, P., Villaseñor-Pineda, L.: Improving Text Classification by Web Corpora. In: *Advances in Soft Computing*, vol. 43, pp. 154–159. Springer, Heidelberg (2007)
5. Hoste, V.: Optimization Issues in Machine Learning of Coreference Resolution. Doctoral Thesis, Faculteit Letteren en Wijsbegeerte, Universiteit Antwerpen (2005)
6. Japkowicz, N.: Learning from Imbalanced Data Sets: A comparison of Various Strategies. In: *AAAI Workshop on Learning from Imbalanced Data Sets*. Tech Rep. WS-00-05, AAAI Press, Menlo Park (2000)
7. Joachims, T.: Transductive inference for text classification using support vector machines. In: *Proceedings of the Sixteenth International Conference on Machine Learning* (1999)
8. Kilgarriff, A., Greffensette, G.: Introduction to the Special Issue on Web as Corpus. *Computational Linguistics* 29(3) (2003)
9. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. *Machine Learning* 39(2/3), 103–134 (2000)
10. Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys* 34(1), 1–47 (2002)
11. Seeger, M.: Learning with labeled and unlabeled data. Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh, Edinburgh, United Kingdom (2001)
12. Solorio, T.: Using unlabeled data to improve classifier accuracy, Master Degree Thesis, Computer Science Department, INAOE, Mexico (2002)
13. Zelikovitz, S., Hirsh, H.: Integrating background knowledge into nearest-Neighbor text classification. In: *Advances in Case-Based Reasoning, ECCBR Proceedings* (2002)
14. Zelikovitz, S., Kogan, M.: Using Web Searches on Important Words to Create Background Sets for LSI Classification. In: 19th International FLAIRS conference, Melbourne Beach, Florida (May 2006)

# A Classifier System for Author Recognition Using Synonym-Based Features

Jonathan H. Clark and Charles J. Hannon

Department of Computer Science, Texas Christian University  
Fort Worth, Texas 76129  
{j.h.clark, c.hannon}@tcu.edu

**Abstract.** The writing style of an author is a phenomenon that computer scientists and stylometrists have modeled in the past with some success. However, due to the complexity and variability of writing styles, simple models often break down when faced with real world data. Thus, current trends in stylometry often employ hundreds of features in building classifier systems. In this paper, we present a novel set of synonym-based features for author recognition. We outline a basic model of how synonyms relate to an author's identity and then build an additional two models refined to meet real world needs. Experiments show strong correlation between the presented metric and the writing style of four authors with the second of the three models outperforming the others. As modern stylometric classifier systems demand increasingly larger feature sets, this new set of synonym-based features will serve to fill this ever-increasing need.

*“The least of things with a meaning is worth more in life  
than the greatest of things without it.”*

**Carl Jung** (1875 - 1961)

## 1 Introduction

The field of stylometry has long sought effective methods by which to model the uniqueness of writing styles. Good models have the quality that they can differentiate between the works of two different authors and label them as such. However, even some of the best models suffer from deficiencies when presented with real world data. This stems from the fact that a writing style is a very complex phenomenon, which can vary both within a literary work and over time [12]. Given these challenges, it is not surprising that the field of stylometry has not yet discovered any single measure that definitely captures all the idiosyncrasies of an author's writings.

Recently, the field of stylometry has moved away from the pursuit of a single “better” metric; modern computational approaches to author recognition combine the power of many features [11, 14]. Thus, the field has begun to recognize that the problem of author recognition is much like a puzzle, requiring the composition of many pieces before the picture becomes clear. In this paper, we present a novel set of synonym-based features, which serves as yet a few more pieces of the much larger puzzle.

Why do we propose a feature set based on synonyms? By examining words in relation to their synonyms, we concern ourselves with the meaning behind those words. For the proposed features, we are primarily interested in answering the question “What alternatives did the author have in encoding a given concept in this language?” In answering this question, we find that we obtain a metric which has a strong correlation with writing style.

## 1.1 Task

The most common application of the techniques discussed in this paper will likely be within a classifier system for author identification. For this task, we are given a set of known authors and samples of literature that are known to correspond to each author. We are then presented with a text sample of unknown authorship and are asked “Of the authors that are known, who is most likely to have written this work?”

## 1.2 Related Work

Some of the earliest features used for author recognition include word length [1, 4], syllables per word [3], and sentence length [8]. Though these measures are found to be insufficient for the case of real world data by Rudman [11], they did make progress in the computational modeling of an author’s writing style. These methods became somewhat more sophisticated with the study of the distinct words in a text by Holmes [6]. Stamatatos et al. present a method that utilizes a vector of 22 features including both syntactic and keyword measures [13]. More recent efforts have gone below the level of the lexicon and examined text at the character-level [7, 10].

The relation of writing style and synonyms is an area that has been much less studied. Coh-metrix, a tool for text analysis based on cohesion calculates measures as polysemy (words having more than one meaning) and hypernymy (words whose meaning is on the same topic but has a broader meaning) [5]. However, these measures were not used for determining what alternative representations of a concept an author had to choose from as is the case in the presented work.

This paper builds on the work of Clark and Hannon [2]. However, this previous work targeted flexibility over accuracy and was evaluated on non-contemporary authors. In this paper, we begin by refining the previous work into a new theoretical framework suitable for combination with other feature sets and present it as model 1. We then present enhancements that cope with the shortcomings of model 1 and compare all 3 models using a more difficult data set.

## 2 Theory

The goal in developing a good model of an author’s writing style is to capture the idiosyncratic features of that author’s work and then leverage these features to match a work of unknown authorship to the identity of its author. As previously stated, a modern system can use hundreds of features at a time. However, each of these features must have a significant correlation with some component of writing style that varies between authors.

We propose that an author’s repeated choice between synonyms represents a feature that correlates with the writing style of an author. Not only do we want to measure which words were selected, but how much choice was really involved in the

selection process. For instance, given the concept of “red,” an author has many choices to make in the English language with regard to exactly which word to select. The language provides many alternatives such as “scarlet” with which an author can show creative expression. More importantly, this creative freedom leads authors to make unique decisions, which can later be used as identifying features. Contrast the example of colors with the word “computer.” It is a concept that maps to relatively few words. Therefore, we might say that an author had less opportunity for expression and that this word is less indicative of authorship.

In the following sections, we present three models, which each represent a point in the natural evolution of this work. Model 1 captures the basic concept of how synonyms relate to an author’s identity while ignoring some of the subtleties of the underlying problem. However, it serves as a conceptual springboard into the more refined models 2 and 3, which perform a deeper analysis of each word to obtain better performance on real world data.

## 2.1 Model 1

Model 1 demonstrates at the most basic level how synonyms can be tied to an author’s identity. Loosely speaking, the idea behind model 1 is that if a word has more synonyms, then the author had more words from which to choose when encoding a given concept. Therefore, the word should be given more weight since it indicates a higher degree of free choice on the part of the author. We model this concept in terms of our task of identification of an unknown author by collecting a feature vector for each word in an author’s vocabulary, running an algorithm over the feature vector, and finding the argument (author) that maximizes the function’s value.

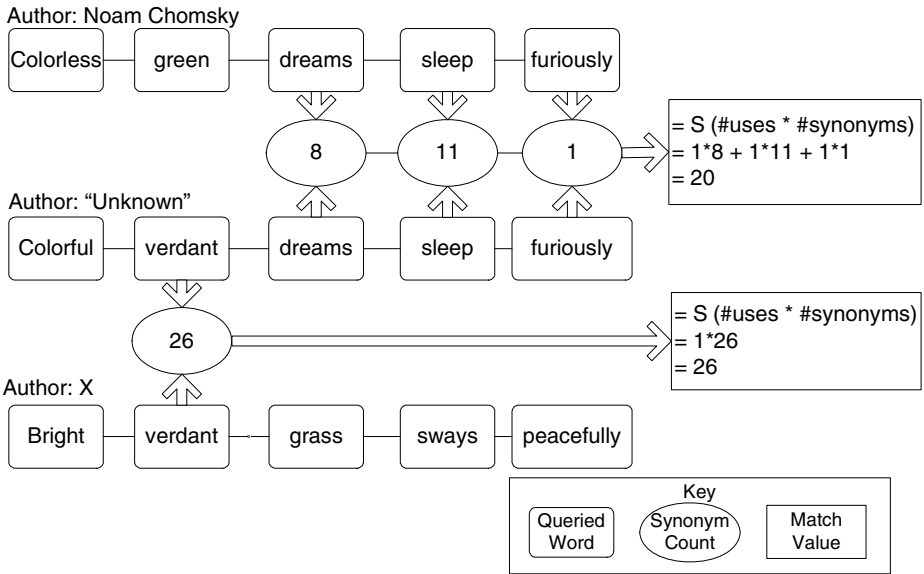
We define the feature vector  $f_l$  of a word  $w$  as having the following elements<sup>1</sup>:

- The number of synonyms  $s$  for  $w$  as according to the WordNet lexical database [9]
- The shared text frequency  $n$  for  $w$ ; that is, if author  $a$  uses word  $w_a$  with frequency  $n_a$  and author  $b$  uses word  $w_b$  with frequency  $n_b$  then the shared frequency  $n = \min(n_a, n_b)$ .

Next we define the function  $match_l$ , which generates an integer value directly related to the stylistic similarity of the unknown author  $u$  with the known author  $k$ :

```
function match1(u, k)
  m ← 0
  for each unique word wu used by author u
    for each unique word wk used by author k
      if wu = wk then
        generate f1 of wu, wk
        m ← m + f1[n] * f1[s]           (see definition of fl above)
      end if
    end for
  end for
  return m
end function match1
```

<sup>1</sup> For clarity, variables peculiar to model 1 are given a subscript of 1.



**Fig. 1.** An example of how match values are calculated for model 1. The top and bottom sentences represent training samples for the authors Noam Chomsky and a hypothetical Author X, respectively. The middle sentence represents an input from an author whose identity is hidden from us. We then perform calculations as shown to determine the author’s identity.

Finally, we define our classifier such that the identity  $I$  of the unknown author is

$$I = \arg \max_{k \in T} \text{match}_1(u, k) \tag{1}$$

where  $T$  is the set of all known authors on which the system was trained.

As a concrete example, consider the above example. (Fig. 1) The words “dreams,” “sleep,” and “furiously” have 8, 11, and 1 synonym, respectively while the word “verdant” has 26 synonyms. A traditional bag-of-words approach would select Noam Chomsky as the author since the sentence of unknown authorship has 3 word matches with Noam Chomsky’s vocabulary. However, model 1 takes into account the fact that the word “verdant” has 26 synonyms and gives it more weight than that of all of the other words in the figure. Thus, model 1 selects Author X as the author of the unknown sentence. Having set forth a simplified model, we now turn to the matter of designing a model robust enough to deal with real world data.

### 2.2 Model 2

In building model 2, we sought to eliminate some of the issues that presented themselves in the implementation and testing of model 1. A careful analysis of the output of model 1 demonstrated two key weaknesses:

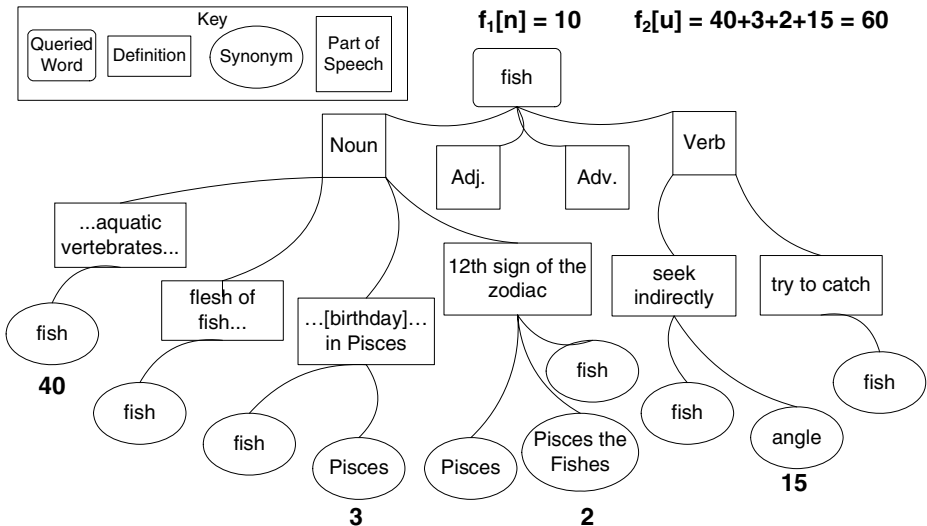
1. A handful of the same high frequency words including pronouns and helping verbs (e.g. “it” or “having”) were consistently the largest contributors

to the value returned by the *match* function even though to a human observer, they are clearly not unique markers of writing style

- Each synonym was being treated as equal although logic suggests that a more common word such as “red” is not as important as an infrequent word such as “scarlet” in determining the identity of an author

To handle the first case in which high frequency words were masking the effect of lower frequency words, we added two improvements over model 1. First, we define a global stopword list that will be ignored in all calculations, a common practice in the field of information retrieval. This reduced the amount of noise being fed to the classifier in the form of words that have lost their value as identifying traits. Second, we revise the function *match* such that we divide the weight for a matched word by the global frequency of that word. The global frequency is computed either via the concatenation of all training data (as is the case for the presented experiments) or via the some large corpus.

In response to the second issue, we see that it is desirable to give words different weights depending on their text frequency. Recall that we seek not only to consider what word choices the author made, but also to consider what the author’s alternative choices were in encoding this concept. Thus, we do not only include the text frequency of the word, but the sum over the global frequencies of all synonyms of each word the author chooses (shown in the example on the following page). Seen in a



**Fig. 2.** An example of a word (*fish*) and its synonyms using the hierarchy defined by WordNet. For sake of discussion, arbitrary weights have been placed under the returned synonyms. These are used to provide context for subsequent examples of models 2 and 3.

different light, we sum the frequencies of all words an author could have chosen for a given concept. In this way, we obtain a value that not only corresponds to the number

of choices the author had, but also how idiomatic those choices are with regard to common language usage.

To summarize, we define the model 2 feature vector  $f_2$  of a word  $w$  as having all elements of  $f_1$  with the following additional elements:

- Whether or not  $w$  is contained in the stop list
- The global frequency  $g$  of  $w$
- The sum  $u$  over the global frequencies of all synonyms of  $w$

The modified version of the function *match*, which we will refer to as *match<sub>2</sub>*, now generates a real value (as opposed to integer) and behaves as follows:

```
function match2(u, k)
  m • 0.0
  for each unique word wu used by author u
    for each unique word wk used by author k
      if wu = wk AND wu, wk is not in stoplist then
        generate f2 of wu, wk
        m • m + f1[n] * f2[u] / f2[g] (see definition of f2 above)
      end if
    end for
  end for
  return m
end function match2
```

To again give a more tangible example of how the model works, we present Fig. 2. Assume that the vocabularies of both the unknown author  $u$  and the known author  $k$  contain the word “fish” and that they used the word 10 and 15 times, respectively. Thus, the word has a shared frequency  $f_1[n]$  of 10. Further, assume that “fish” occurred 20 times in some large corpus from which we obtain the global frequency. Since “fish” is not a stop word, it will be given a non-zero weight. Also note that fish has four unique synonyms with global frequencies of 40, 3, 2, and 15, respectively. Thus, the sum over the global frequencies of the synonyms  $u$  is 60. With this information we can now calculate the value of  $m$  as shown in the function *match<sub>2</sub>* by  $10 * 60 / 20 = 30$ .

The additional features in model 2 make it much more robust than model 1. It considers not only the number of alternative choices an author had, but how idiomatic those choices are with regard to how language is commonly used. We now look toward model 3, which attempts to incorporate still more linguistic information into the synonym-based feature set.

### 2.3 Model 3

In model 3, we attempt to exploit the morphology of the English language. Though English is not considered a morphologically rich language, it certainly does have cases in which the morphology causes what the average speaker might consider the same word to be mapped to two different words (e.g. “give” and “gives”).

Model 3 attempts to compensate for this phenomenon by applying stemming to each word in the author’s vocabulary. This process of stemming is the only change between models 2 and 3. The assumption here is that it is not important which



morphological form of a word an author chooses. Rather, in model 3, we place the emphasis on which synonym and which shade of meaning an author chooses to represent a given concept. We leave it up to the results to indicate whether or not this is a meaningful assumption.

### 3 Implementation

#### 3.1 Corpus

To perform the author identification task, we selected a corpus consisting of 1,333,355 words from four authors including Jacob Abbott, Lydia Child, Catharine Traill, and Charles Upham. To ensure our system was not using stylistic markers of time periods in differentiating between authors, the authors were selected such that they were all born within roughly a year of each other (1802 – 1803). All works used in the test set were retrieved from Project Gutenberg<sup>2</sup> and are freely available for download. After obtaining the data, we removed all portions of the text that would not be considered an author's original work (i.e. tables of contents, prefaces, etc.). The remaining body of text was then divided evenly into five folds, one of which was used as training data and the other four being left as test cases. Basic statistics for the corpus are presented in Table 1.

**Table 1.** This table shows word counts for each fold of the 1,333,355 word corpus

Author	Total Words		Unique Words	
	Testing (Avg)	Training	Testing (Avg)	Training
Abbott	60,316	57,898	4,763	6,198
Child	87,187	90,960	7,646	6,963
Traill	59,713	63,482	6,576	7,168
Upham	57,987	57,075	6,297	6,858

#### 3.2 WordNet

One very important tool in implementing the system was Princeton WordNet<sup>3</sup> [9]. WordNet is a lexical database of the English language that has qualities similar to both a dictionary and a thesaurus. Most importantly, it contains links between synonyms which may be traversed as “synsets.” For example, Fig. 2 shows a synset taken from WordNet. Version 2.1 of WordNet, used in this research, contains 207,016 word-sense pairs within 117,597 synsets. WordNet also includes a very simple yet effective morphological processor called Morphy, which we used to perform stemming for model 3.

<sup>2</sup> Project Gutenberg is accessible at <http://www.gutenberg.org>

<sup>3</sup> The can be downloaded at <http://wordnet.princeton.edu/>

### 3.3 Stop Word List

To prevent conflict of interest, we used a stop word list from an external source, the Glasgow University Information Retrieval group<sup>4</sup>. The list contained 319 of the most common words in the English language. At runtime, we used the WordNet Morphy morphological processor to stem the words on the Glasgow stop list to obtain more stop words. Finally, we augmented this list with names from the U.S. Census Bureau website, which included the most frequent 90% of both first and last names, as indicated by the 1990 census.<sup>5</sup> The combination of all these sources was used as the stop word list for models 2 and 3.

### 3.4 Pre-processing

Incident to using WordNet, part of speech tagging is recommended so that WordNet can narrow down which senses of the word might be intended (see Fig. 2). For this purpose, we employed the Stanford Log-Linear Part of Speech Tagger<sup>6</sup> [15]. The supplied trained tagger was used as there was no compelling reason for custom training.

## 4 Results

Results for each section are presented for the three cases of classifying between 2, 3, or 4 authors at a time. For all cases, all 4 test folds of each author were evaluated against some number of trained models. In the case of classifying between 3 authors at a time, all possible  ${}_4C_3$  (4) combinations of 3 authors were evaluated and results were then averaged over these sets. Similarly, for the case of classifying between 2 authors at a time, all  ${}_4C_2$  (6) combinations were tested. Results are reported as precision, recall, and F1 scores. Precision is defined as the number of test cases (i.e. folds) correctly reported as being written by a given author divided by the total number of test cases reported as being written by that author. Similarly, recall is defined as the number of test cases correctly reported divided by the total number of correct test cases possible. Finally, the F1 score is calculated as the harmonic mean of precision and recall.

### 4.1 Model 1

We begin by analyzing the performance of model 1. Of the three models, model 1 was produced the lower overall F1 scores (see Table 2). For the case of differentiating between two authors at a time, model 1 produced better than chance results. As model 1 has to deal with choosing between more authors, performance declines steeply. Certainly we prefer a model that displays both higher accuracy and more graceful degradation when faced with larger numbers of authors. To realize these characteristics, we turn to models 2 and 3.

---

<sup>4</sup> This stop word list is located at [http://www.dcs.gla.ac.uk/idom/ir\\_resources/linguistic\\_utils/](http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/)

<sup>5</sup> The name list is available at <http://www.census.gov/genealogy/www/freqnames.html>

<sup>6</sup> The tagger may be obtained at <http://nlp.stanford.edu/software/tagger.shtml>

**Table 2.** Precision, recall, and F1 scores for model 1

Author	Authors = 4			Authors = 3			Authors = 2		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Abbott	0.000	0.000	0.000	0.000	0.000	0.000	0.333	1.000	0.500
Child	1.000	0.267	0.421	1.000	0.353	0.522	1.000	0.522	0.686
Traill	0.250	1.000	0.400	0.500	0.462	0.480	0.750	0.563	0.643
Upham	0.000	0.000	0.000	0.083	1.000	0.154	0.417	1.000	0.588
Overall	0.313	0.313	<b>0.313</b>	0.396	0.396	<b>0.396</b>	0.625	0.625	<b>0.625</b>

## 4.2 Model 2

Model 2 exhibited the most desirable qualities of all the models evaluated. Not only was it highly accurate in terms of F1 score, but it also displayed a graceful degradation curve as it was faced with discerning between larger numbers of authors. The benefits of having probed more deeply into the frequency of all of a word's synonyms and utilizing global frequencies in our feature vector are underlined by these results (see Table 3).

**Table 3.** Precision, recall, and F1 scores for models 2 and 3

Author	Authors = 4			Authors = 3			Authors = 2		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Abbott	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Child	1.000	0.080	0.889	1.000	0.857	0.923	1.000	0.923	0.960
Traill	0.750	1.000	0.857	0.833	1.000	0.909	0.917	1.000	0.957
Upham	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Overall	0.938	0.938	<b>0.938</b>	0.958	0.958	<b>0.958</b>	0.979	0.979	<b>0.979</b>

## 4.3 Model 3

Having performed the additional step of stemming for model 3, the expected result was that scores would increase. In actuality, there was no change from the scores of model 2 (Table 3). To clarify the meaning of these results, we also calculated the average percent difference between the weights returned by the *match* function for the top two authors (Table 4). This gives us a rough estimate of how “confident” the system was in making its choice with a larger percentage difference being more desirable. For all cases, model 2 produced these larger differences between its top 2 matches. Thus, we conclude not only that we received no benefit from stemming, but that it had a negative effect on the output, be it very small negative effect. From this, we draw that the author's choice about which form of a word to use is an important choice and should not be discarded via stemming.

**Table 4.** This table shows the percent difference between the weights returned by the *match* function for the top two authors, averaged over all test cases

	Authors = 4		Authors = 3		Authors = 2	
	Model 2	Model 3	Model 2	Model 3	Model 2	Model 3
Abbott	0.136	0.051	0.137	0.077	0.157	0.125
Child	0.120	0.160	0.150	0.188	0.204	0.249
Traill	0.146	0.104	0.168	0.125	0.218	0.179
Upham	0.144	0.061	0.196	0.083	0.265	0.127
Overall	<b>0.135</b>	0.098	<b>0.164</b>	0.121	<b>0.211</b>	0.172

## 5 Conclusion

We have presented a novel set of synonym-based features for use in a classifier system that performs author identification. As evidenced in the results, these features perform well on real world data when properly tuned (i.e. models 2 and 3). However, to harness the full potential of this feature set, it should be combined with many other features so that a full range of characteristics of writing style are considered. This new set of synonym-based features provides yet another tool with which stylometric classifier systems will be able to analyze written language.

## References

1. Brinegar, C.S.: Mark Twain and the Quintus Curtius Snodgrass Letters: A Statistical Test of Authorship. *Journal of the American Statistical Association* 58 (1963)
2. Clark, J.H., Hannon, C.J.: An Algorithm for Identifying Authors Using Synonyms. In: ENC 2007 (2007)
3. Fucks, W.: On the mathematical analysis of style. *Biometrika* 39, 122–129 (1952)
4. Glover, A., Hirst, G. (eds.): *Detecting stylistic inconsistencies in collaborative writing*. Springer, London (1996)
5. Graesser, A.C., McNamara, D.S., Louwerse, M.M., Cai, Z.: Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, and Computers* 36, 193–202 (2004)
6. Holmes, D.I.: Authorship attribution. *Computers and the Humanities* 28 (1994)
7. Khmelev, D.V., Tweedie, F.J.: Using Markov Chains for Identification of Writers. *Literary and Linguistic Computing* 16, 299–307 (2002)
8. Mannion, D., Dixon, P.: Sentence-length and Authorship Attribution: the Case of Oliver Goldsmith. *Literary and Linguistic Computing* 19, 497–508 (2004)
9. Miller, G.A.: WordNet: A Lexical Database for English. *Communications of the ACM* 38, 39–41 (1995)
10. Peng, F., Schuurmans, D., Keselj, V., Wang, S.: Language Independent Authorship Attribution using Character Level Language Models. In: 11th Conference of the European Chapter of the Association for Computational Linguistics (2004)
11. Rudman, J.: The State of Authorship Attribution Studies: Some Problems and Solutions. *Computers and the Humanities* 31, 351–365 (1998)

12. Smith, J.A., Kelly, C.: Stylistic constancy and change across literary corpora: Using measures of lexical richness to date works. *Computers and the Humanities* 36, 411–430 (2002)
13. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Automatic Text Categorization in Terms of Genre and Author. *Computational Linguistics* 26, 471–495 (2000)
14. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Computer-Based Authorship Attribution Without Lexical Measures. *Computers and the Humanities* 35 (2001)
15. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *HLT-NAACL*, pp. 252–259 (2003)

# Variants of Tree Kernels for XML Documents

Peter Geibel, Helmar Gust, and Kai-Uwe Kühnberger

University of Osnabrück, Institute of Cognitive Science, AI Group, Germany  
{pgeibel,hgust,kkuehnbe}@uos.de

**Abstract.** In this paper, we discuss tree kernels that can be applied for the classification of XML documents based on their DOM trees. DOM trees are ordered trees, in which every node might be labeled by a vector of attributes including its XML tag and the textual content. We describe four new kernels suitable for this kind of trees: a tree kernel derived from the well-known parse tree kernel, the set tree kernel that allows permutations of children, the string tree kernel being an extension of the so-called partial tree kernel, and the soft tree kernel, which is based on the set tree kernel and takes into account a “fuzzy” comparison of child positions. We present first results on an artificial data set, a corpus of newspaper articles, for which we want to determine the type (genre) of an article based on its structure alone, and the well-known SUSANNE corpus.

## 1 Introduction

One of the main problems considered in text mining is the classification of documents. In the emerging new field of so-called web corpora one is interested in relatively complex concepts, as for instance, the genre or type a document [1]. For determining the genre of a text, not only the occurring words play a role, but it is to a large extent determined by its visual and organizational structure. In this article, we will therefore investigate the structure based classification of XML documents based on their DOM trees (Document Object Model).

Kernel methods like the SVM [2] can be applied to non-vectorial data like sequences, trees, and graphs by defining an appropriate kernel for the data at hand [3]. This can either be accomplished by directly defining an appropriate function that is positive-semidefinite (PSD, e.g., [4]) or by explicitly defining an appropriate feature mapping for the structures considered. An example of a kernel for structures is the *parse tree kernel* [5,6], which is applicable to parse trees of sentences. In contrast to parse trees, in which a grammar rule applied to a non-terminal determines number, type and sequence of the children, structural parts of a text represented by its DOM tree might have been deleted, permuted or inserted compared to a text considered similar. This higher flexibility should be taken into account in the similarity measure represented by the tree kernel, because otherwise the value for similar documents might be unreasonably small. As a second difference, DOM trees are often much more complex than parse trees so that complexity issues play a rule in practical applications.

In this paper, we extend previous work on tree kernels suitable for XML data in several respects. Our four kernels are based the *simple labeled ordered tree kernel* (SLOTK), which itself is an extension of the parse tree kernel to trees not generated by a grammar. It can be shown that the SLOTK is a so-called convolution kernel [7]. The proof of this fact implies the positive definiteness of our DOM tree kernels.

Based on the SLOTK, which is not really suited for DOM trees, we present new kernels that are useful in the context of HTML and XML documents. The *LeftTK* is a straightforward generalization of the labeled ordered tree kernel to DOM trees. The *set tree kernel* (SetTK) allows permutations of child subtrees in order to model document similarity more appropriately, but can still be computed relatively efficiently.

In this article, we suggest a straightforward method for including node properties in a natural way by combining the respective tree kernel with kernels operating on node properties. Based on this technique, the *soft tree kernel* (SoftTK) combines the set tree kernel with a fuzzified comparison of child positions in order to account for the ordering of subtrees to some extent while at the same time still allowing permutations of subtrees.

Both Kashima and Konayagi [8] and Moschitti [9] presented kernels for labeled ordered trees. Both kernels are based on the idea of employing a string kernel for the sequence of children of a tree node. We present a (slight) extension of this idea in allowing for the inclusion of complex node properties and using an alternative computation: the *string tree kernel* (StringTK) is derived from a combination of the simple labeled ordered tree kernel with the well-known string kernel defined in [10], which differs from the computations in [8] and [9].

The rest of this paper is structured as follows. After a short definition of trees and various kinds of subtrees in Section 2, we will describe the parse tree kernel, and afterwards the simple labeled ordered tree kernel. The new kernels are defined in section 3, followed by an experimental evaluation in section 4. We consider a small artificial dataset, the prediction of the type of newspaper articles based on their structure only, and results for the SUSANNE corpus. A short conclusion can be found in section 5.

## 2 The Parse Tree Kernel

In the following, we consider labeled, ordered, rooted trees whose nodes  $v \in V$  are labeled by a function  $\alpha : V \rightarrow \Sigma$ , where  $\Sigma$  is a set of node labels. The elements of  $\Sigma$  can be thought of as tuples describing the XML tag and attributes of a non-leaf node in the DOM tree. Leaves are usually labeled with words or parts of texts. We will incorporate node information by using a kernel  $k^\Sigma$  operating on pairs of node labels, i.e., on tags, attributes, and/or texts. Two trees  $T$  and  $T'$  are called isomorphic if there is a bijective mapping of the nodes that respects the structure of the edges, the labellings specified by  $\alpha$  and  $\alpha'$ , and the ordering of the nodes.

Collins and Duffy [5] defined a tree kernel that can be applied in the case of parse trees of natural language sentences (see also [6]), in which non-leaf nodes are labeled with the non-terminal of the node, and leaves with words. The production applied to a non-leaf node determines the number, type, and ordering of the child nodes. Kernels in general, and the parse tree kernel in particular, can be defined by specifying a sequence of features which are pattern trees in the case of the parse tree kernel. Collins and Duffy tailored their kernel to parse trees meaning that they only consider patterns trees  $t$  corresponding to incomplete parse trees, in which some productions have not yet been applied. This means that if a non-terminal has been expanded, all of its children have to be present. Such trees are called *partial parse trees* in the following. Collins and Duffy excluded trees consisting of a single node only from the sequence of pattern trees.

For a tree  $T$  and a feature or pattern tree  $t$  in the tree sequence, we define  $\phi_t(T)$  as the number of subtrees of  $T$  that correspond to partial parse trees and are *isomorphic* to the feature tree  $t$ . Let  $\phi(T)$  denote the arbitrarily ordered sequence of all feature values, i.e., numbers, obtained by this procedure. Based on this sequence, the parse tree kernel is defined as  $k(T, T') = \langle \phi(T), \phi(T') \rangle = \sum_t \phi_t(T) \cdot \phi_t(T')$ , which is guaranteed to be finite, because there can only be a finite number of partial parse trees  $t$  common in  $T$  and  $T'$ .

Collins and Duffy showed that  $k(T, T')$  can be computed efficiently by determining the number of possible *mappings* of isomorphic partial parse trees (excluding such consisting of a single node only). For nodes  $v \in V$  and  $v' \in V'$ , the function  $\Delta(v, v')$  is defined as the number of isomorphic mappings of partial parse trees rooted in  $v$  and  $v'$ , respectively. Based on  $\Delta$ , the kernel value can be computed as

$$k(T, T') = \sum_{v \in V, v' \in V'} \Delta(v, v'). \tag{1}$$

The computational complexity is  $O(|V| \cdot |V'|)$ .

The  $\Delta$ -function can be computed recursively by setting  $\Delta(v, v') = 0$  for *any* words and if the productions applied in  $v$  and  $v'$  are different. If the productions in  $v$  and  $v'$  are identical and both nodes are pre-terminals, we set  $\Delta(v, v') = 1$ . Pre-terminals are non-terminals occurring directly before leaves corresponding to words. Identical productions in pre-terminals imply identical words. For other non-terminals with identical productions, Collins and Duffy use the recursive definition  $\Delta(v, v') = \prod_{i=1}^{n(v)} (1 + \Delta(v_i, v'_i))$ , where  $v_i$  is the  $i$ -th child of  $v$ , and  $v'_i$  is the  $i$ -th child of  $v'$ .  $n(v)$  denotes the number of children of  $v$  (here corresponding to that of  $v'$ ). It is possible to weight deeper trees using a factor  $\lambda \geq 0$ .

In order to apply this technique to DOM trees, we have to modify the definition in a first step. For instance, there are no pre-terminals in DOM trees, and the leaves are not necessarily words. Moreover, we want to include the similarity of node labels. We can both simplify and generalize parse tree kernels to arbitrary labeled, ordered trees using the following definition of a modified  $\Delta$ -function that can be used with the definition in eq. (1).  $\Delta_{\text{SLOTK}}$  operates on arbitrary trees and is defined as follows. If there are either no children, or the number of children differs we set



$$\Delta_{\text{SLOTK}}(v, v') = \lambda \cdot k^\Sigma(\alpha(v), \alpha(v')).$$

For non-leaves with the same number of children  $n(v)$ , we set

$$\Delta_{\text{SLOTK}}(v, v') = \lambda \cdot k^\Sigma(\alpha(v), \alpha(v'))(1 + \prod_{i=1}^{n(v)} \Delta_{\text{SLOTK}}(v_i, v'_i)). \tag{2}$$

Compared to the recursion of the parse tree kernel given above, the number “1” now appears in front of the product because we no longer exclude pattern trees consisting of a single node only. The SLOTK could, for instance, be applied to *logical* terms and will form the basis for the DOM tree kernels, in which we want to include common properties of child trees of two nodes  $v$  and  $v'$ , even if  $n(v) \neq n'(v')$ , which is not possible with the SLOTK.

It can be shown that the SLOTK is a so-called convolution kernel, which was introduced by Haussler [7]. A convolution kernel operates on a pair of structures. Its value is computed by looking at possible decompositions of each structure into  $n$ -dimensional parts, for which a “sub-kernel” has to be defined. For the SLOTK, such a part is the induced subtree rooted in a node, i.e.,  $n = 1$ .

Although it is easy to show that  $k$  is a convolution kernel, given that  $\Delta_{\text{SLOTK}}$  is a kernel, it is much more tricky to show that  $\Delta_{\text{SLOTK}}$  is one. It can be shown, however, that  $\Delta_{\text{SLOTK}}$  is not a convolution kernel itself, but based on a convolution kernel corresponding to the product occurring in (2). The proof of the kernel property of  $\Delta_{\text{SLOTK}}$  is a bit tricky, because the two cases for leaves and non-leaves have to be collapsed into a single definition. A second problem arises from the fact, that there is no common branching factor for all nodes (corresponding to the fixed  $n$  of the convolution kernel). The proof uses induction on the structural complexity of the trees involved. It is quite lengthy and technical, so we did not include its details.

It follows from the proof that it is possible to just multiply with  $k^\Sigma(\alpha(v), \alpha(v'))$ , which is difficult to see from a feature space interpretation. Kashima and Konayagi [8], for instance, present a technique for including node attributes that involves computations of  $\sum_a k^\Sigma(\alpha(v), a)k^\Sigma(a, \alpha'(v'))$  where  $a$  ranges over all possible nodes label. This is obviously only possible if  $\Sigma$  is finite. In this respect, our tree kernels extend previous approaches.

It also follows that we might replace the product in (2) with any kernel operating on the child tree sequences without losing positive-definiteness. Based on this, we can now define the DOM tree kernels.

### 3 Kernels for DOM trees

We will describe tree kernels tailored for XML documents in the following section. We usually only give the recursive case that the kernel is applied to a pair of trees, where at least one is no leaf. The case of two leaves is identical to the first case in the SLOTK-definition given above, i.e., the kernel value is  $\lambda$ .

### 3.1 The Left-Aligned Tree Kernel (LeftTK)

The LeftTK is relatively straightforward extension of the SLOTK defined in section 3. Its basic idea is to compare just as many children as possible using the given order  $\leq$  on the child nodes if the number of children differs. If we choose  $k^\Sigma$  as the identity kernel we can arrive at a feature space interpretation by allowing arbitrary trees  $t$  in the sequence of pattern trees. For defining the feature value  $\phi_t(T)$ , a feature tree  $t$  is allowed to be a general subtree of a tree  $T$ , with the restriction that only the rightmost children of a node may be missing (possibly all), i.e., the subtree occurs left-aligned.

Note that when comparing two trees  $T$  and  $T'$ , we have to take into account shorter prefixes of the child tree sequences of two nodes  $v$  and  $v'$  as well. This is done by redefining the recursive part of the  $\Delta$ -function (2) by

$$\Delta(v, v') = \lambda \cdot k^\Sigma(\alpha(v), \alpha(v')) \left( 1 + \sum_{k=1}^{\min(n(v), n'(v'))} \prod_{i=1}^k \Delta(v_i, v'_i) \right). \tag{3}$$

Note that this way trees occurring more to the left have a higher influence on the kernel value than trees that occur more to the right, which might be a good property for some classification tasks on texts.

### 3.2 The Set Tree Kernel (SetTK)

The DOM tree kernel defined in the previous section does not allow the child trees of  $v$  and  $v'$  to be permuted without a high loss in similarity as measured by the kernel value  $k(T, T')$ . This behavior can be improved, however, by considering the child tree sequences as *sets* and applying a so-called set kernel to them, which is also an instance of the convolution kernel. The corresponding definition of  $\Delta$  in (2) is obtained with

$$\Delta(v, v') = \lambda \cdot k^\Sigma(\alpha(v), \alpha(v')) \left( 1 + \sum_{i=1}^{n(v)} \sum_{i'=1}^{n'(v')} \Delta(v_i, v'_{i'}) \right), \tag{4}$$

i.e., all possible pairwise combinations of child trees are considered<sup>1</sup>. Note that the computational complexity now is  $O(b^2 \cdot |V| \cdot |V'|)$  where  $b$  is the maximum branching factor of the tree. The complexity of computing the LeftTK only depends on  $b$ .

When looking for a suitable feature space in the case  $\lambda = 1$  and  $k^\Sigma = k^{id}$ , we find that the definition in (4) corresponds to considering *paths* from the root to the leaves. This is a well-known technique for characterizing labeled graphs (see, e.g., [3,11]), which can also be applied to trees.

Using only label sequences instead of trees is known to potentially result in a loss of structural information because we loose information on how the single paths characterizing a tree  $T$  are related to each other. This can, however, be repaired with the kernel defined in the following section.

---

<sup>1</sup> It is possible to divide the product by  $n(v)n'(v')$ .

### 3.3 The Soft Tree Kernel (SoftTK)

The basic idea of the soft tree kernel is to take the position of a node in the child sequence into account. The position  $\mu(v_i) = i$  of some child of a node  $v$  can be used as an attribute of the respective node. The position of the root node is defined as 1. When comparing two positions we are interested in their distance. This suggests to use of the RBF kernel defined as  $k_\gamma(x, y) = e^{-\gamma(x-y)^2}$  for node positions  $x$  and  $y$ . The maximum value is attained for  $x = y$ .  $\gamma$  is a parameter to be set by the user. It determines how different the positions are allowed to be. The comparison of positions can be seen as being as soft or “fuzzy”. We can state the recursive part of the definition of the soft tree kernel as

$$\Delta(v, v') = \lambda \cdot k^\Sigma(\alpha(v), \alpha(v')) \cdot k_\gamma(\mu(v), \mu'(v')) \cdot \left(1 + \sum_{i=1}^{n(v)} \sum_{i'=1}^{n'(v')} \Delta(v_i, v'_{i'})\right).$$

Giving a feature space interpretation in terms of pattern trees is again difficult, because of the use of  $k^\Sigma$  and  $k_\gamma$ . The string tree kernel defined in the next section provides an alternative method for taking into account the ordering of the children of a node.

### 3.4 The String Tree Kernel

Alessandro Moschitti [9] describes an extension of the parse tree kernel called the partial tree kernel. The partial tree kernel is based on string kernels, i.e., one considers common subsequences of the two child tree sequences with gaps allowed. In the following, we will present a slight extension of the partial tree kernel that is based on a different computation of the string kernel and includes node attributes. Moreover, we do not consider all subsequences, but allow the user to specify a maximum length in order to limit computational complexity.

To define the  $\Delta$ -function formally, we consider index sequences  $I$  which are finite, ascending sequences of numbers  $\geq 1$ . The sequence  $I$  characterizes a subsequence of the sequence of child trees of a node.  $|I|$  denotes the length of the sequence, i.e., its number of entries. Let  $l(I) = |I| - I_1$  be the length of the subsequence characterized by  $I$  (note the difference to  $|I|$  which refers to the number of entries in  $I$ ). We introduce a new parameter  $\rho \in [0; 1]$  that is used for penalizing the length of a sequence. The parameter  $\lambda$  is used as above.

In contrast to the partial tree kernel, this kernel will be called *string tree kernel* in the following. In order to be able to control the computational complexity, we introduce a parameter,  $L$ , for the maximum substring length considered resulting in the definition. We obtain  $\Delta$  from (2) by replacing the product with

$$P(v, v') = \sum_{p=1}^L \overbrace{\sum_{I, I', |I|=|I'|=p, I_p \leq n(v), I'_p \leq n'(v')}^{D_p(v_1 \dots v_{n(v)}, v'_1 \dots v'_{n'(v)})} \rho^{l(I)+l(I')} \prod_{i=1}^{|I|} \Delta(v_{I_i}, v'_{I'_i})$$

for any pair  $v$  and  $v'$ .

Class 1:	Class 2:
f(n,h(m),g(a,b(e),c))	f(n,h(n),g(c,m,b,n,e(a)))
f(h(m,g(a,b(e),c)))	f(h(h(m),n,b),g(c,b,n,m,e(a)))
f(g(a,b(e),c,n),h(m))	f(h(g(h(n),c,b,e(a))),b)
f(g(a,b(e),c,n,m))	f(h(g(n,c,b,e(a))),h(m,b,h(a,n)))
f(a,m,m,h(h(g(a,b(e),c))),n)	f(g(b(e),c,a),g(c,n,b,e(a)))
f(g(a,b(e),c),g(e(a),c,a))	f(g(c,h(c,n),b,h(h(h(b))),e(a)))
f(h(m,c),g(a,b(e),c),g(m,n,m))	f(g(c,b,e(a)),g(m,b,n,a,b(e),n,c))
f(g(a,b(e),c,h(h(m),n,b)))	f(b,g(c,b,e(h(b,m,a),a)))
f(h(h(g(a,b(e),c))))	f(g(g(a),c,d,m),g(c,b,h(n),e(a)))
f(g(a,b(e),c),a)	f(g(m,c,b,e(a),h(a)))
Class 3:	Class 3 (continued):
f(n,g(c,n,b,m,a(e)),h(m))	f(g(h(m,e),a(e),h(b),c,a,b))
f(m,e,b,h(g(b,n,c,m,a(e))))	f(g(b,h(m),c,h(n),a(b,e,a)))
f(h(h(a(e)),m,b,n,n,c),b,e)	f(g(m,h(n,h(n)),g(b,e,c,a(e))))
f(e,b,g(m,c,a,b,e,e,a(e)))	f(g(a(e),g(h(n)),b,c),b,n)
f(b,e,g(a(e),m,b,h(a),m,c))	f(h(h(e),h(n),b,h(n),m),g(c,m,n,b,a(e)))

**Fig. 1.** A small training set with three classes

Note that the function  $P$  resembles the standard definition of a string kernel [10,9]. For two nodes  $v$  and  $v'$ , the value  $P(v, v')$  corresponds to the number of mappings of isomorphic subtree sequences with a length of at most  $L$ .  $P$  will be computed by computing the number of isomorphic mappings of child tree sequences of length  $p$ ,  $D_p(v_1 \dots v_n(v), v'_1 \dots v'_{n'}(v'))$ , which is determined more efficiently by the recursive definition given below. In contrast to the string kernel defined by Lodhi et al., [10] we have to take the fact into account that a “symbol” of the child tree sequence corresponds to a whole tree. Hence we have to compute possible mappings of common subtrees.

We define  $sv$  to stand for a child tree sequence with last element  $v$ . We then have

$$D_p(sv, s') = D_p(s, s') + \sum_{j, \Delta(v, v'_j) > 0} \Delta(v, v'_j) \cdot D'_{p-1}(s, v'_1, \dots, v'_j) \rho^2 \quad (5)$$

We set  $D_p(s, s') = 0$  if  $\min(|s|, |s'|) < p$  holds. Note that we had to extend the string kernel from [10] with  $\Delta(v, v'_j)$ .

The function  $D'_p$  is similar to  $D_p$  except for a different penalty for gaps. It can be obtained from [5] by replacing  $D_p(s, s')$  with  $D'_p(s, s')$  and  $\rho^2$  with  $\rho^{|s'| - j + 2}$ . We set  $D'_p(s, s') = 0$  if  $\min(|s|, |s'|) < p$  holds, and  $D'_0(s, s') = 1$ .

Compared to the string kernel [10] the usage of  $\Delta(v, v'_j)$  is new. For a string of symbols,  $\Delta(v, v'_j)$  can be thought to amount to one if  $v$  and  $v'_j$  are identical symbols. Note that the complexity has to be multiplied with  $L$ , i.e., the length of substrings considered.

Kashima and Koyanagi [8] give a definition of a kernel for semi-structured data which is similar to the partial tree kernel and the string tree kernel, although

**Table 1.** Optimal F-Measures (Leave-One-Out)

	Class 1	Class 2	Class 3
TagTK	0.727	0.6	0.736
LeftTK	0.909	0.363	0.44
SetTK	0.952	1.00	1.00
SoftTK	1.0	1.0	1.0
StringTK	1.0	1.0	1.0

their recursive computation differs from our definition, and they also do not allow to use penalties for gaps.

The DOM tree kernel defined in section 3.1 and the set tree kernel from section 3.2 can be seen as restricted forms (but not special cases) of the partial tree kernel that are obtained when imposing additional restrictions on the allowed index sequences  $I$  and  $I'$ . The dom tree kernel only considers sequences  $I$  and  $I'$  defined as (1), (1, 2), (1, 2, 3), and so on, whereas the set tree kernels allows different index sequences for  $I$  and  $I'$ , but their lengths are restricted to 1.

## 4 Experiments

In order to validate our kernels, we applied them to a small training set with three classes and 10 examples in each class, resulting in 30 artificial trees that are depicted in Fig. 1. The classes have been defined as follows:

1. The class 1 examples all have a left-aligned subtree of the form  $g(a, b(e), c)$ .
2. The class 2 examples all have a general ordered subtree of the form  $g(c, b, e(a))$ , where gaps are allowed but the ordering of the subtrees  $c$ ,  $b$  and  $e(a)$  has to be preserved.
3. The class 3 examples contain subtrees of the form  $g(c, b, a(e))$ , where the child trees  $c$ ,  $b$  and  $a(e)$  are allowed to occur reordered and gaps might have been inserted, too.

We compared the four kernels LeftTK, SetTK, SoftTK, and StringTK with the baseline approach of classifying the trees according to the occurring tags (TagTK). We extended the LIBSVM [12] with implementations of our kernels. The optimal F-Measures<sup>2</sup> for the different approaches and the three classes can be found in Table 1.

The implementations of the SoftTK and the StringTK performs best on our small dataset: they achieve an optimal F-measure of 1.00 for each class. SetTK performs a bit worse for class 1 with an F-measure of 0.952. Although SoftTK and StringTK perform identically for our small dataset, SoftTK can be computed more efficiently, for its asymptotic worst-case complexity does not depend on the length of string considered, which is the case for the StringTK. Note that the

<sup>2</sup>  $\frac{2rp}{r+p}$  where  $r$  is the recall of the respective class, and  $p$  the precision.

**Table 2.** Results for Susanne Corpus: F-Measures

Cat.	TagTK	-N	LeftTK	-N	SetTK	-N	SoftTK	-N	StringTK	-N
A	0.968	0.97	0.538	0.4	0.97	0.968	1.0	1.0	1.0	1.0
G	0.75	0.688	0.405	0.367	0.733	0.89	0.733	0.89	0.687	0.727
J	0.903	0.903	0.688	NAN	0.903	0.903	0.903	0.953	0.8	0.903
N	0.968	0.968	0.4	NAN	0.967	1.0	0.967	1.0	0.967	0.967

computational complexity of both kernels, however, is quadratic in the number of nodes of the trees, and also in the maximum branching factor. The LeftTK, our simplest approach, performed the worst for classes 2 and 3 – even worse than the default approach given by TagTK. It seems to be useful, however, for class 1, because it is defined by a certain left-aligned subtree. It should be noted that LeftTK can be computed more efficiently than SetTK, SoftTK, and StringTK, because its complexity only depends linearly on the maximum branching factor of the trees.

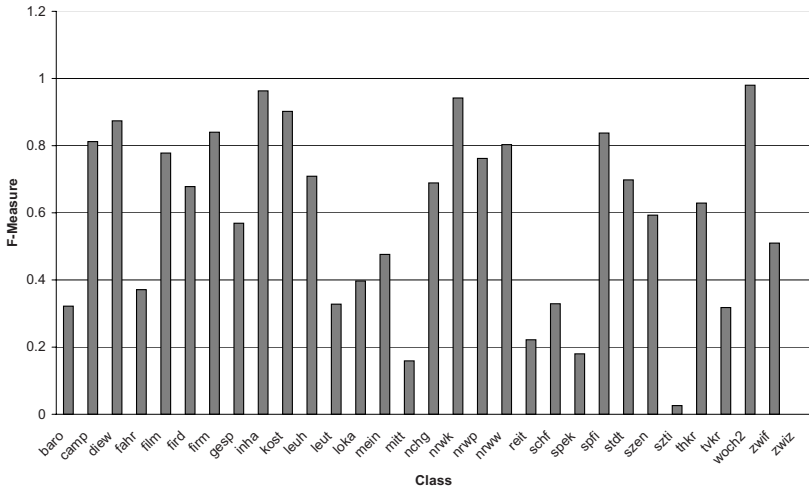
The SUSANNE Corpus [13] consists of 64 files which are annotated version of texts from the Brown corpus, each of which contains more than 2000 words. Sixteen texts are drawn from each of the following genre categories: A (press reportage), G (belles lettres, biography, memoirs), J (learned, mainly scientific and technical, writing), N (adventure and Western fiction). Because we are interested in the structure based classification, all information on specific words were removed. We also kept only a simplified version of the parsed text and sentence structure, with tags like *N* (noun phrase), *V* (verb phrase), etc., in which the specific tag *Y* was used to denote the occurrence of **some** word. Interestingly enough, we could still obtain very good classification results, see table 2.

In Table 2, the estimated optimal F-measure for the four categories are shown (using the leave-one-out method). For every type of kernel (see above), we considered the original definition and the normalized version given by  $k'(x, y) =$

$$\frac{k(x, y)}{\sqrt{k(x, x)k(y, y)}}.$$

It can be seen, that the classes A, J, N can be learned quite well with any of the approaches, except LeftTK. For class G, however, normalized SetTK and SoftTK perform best, and particularly much better than the baseline approach TagTK that only looks at occurring symbols. Note that for large values of  $\gamma$ , SoftTK behaves like SetTK.

We also performed first experiments on a corpus containing DOM trees of approx 35000 newspaper articles to be classified according to their type as, e.g., “chronicle of the week” etc. (cf. [14]). Each article was described by its structure related to sections, paragraphs and sentences. Normal text and headlines could be distinguished by the usage of different tags. However, we did not include information about the words and sentence structure, because we wanted to see how well we can determine the genre of each article based on its structure only. For the 31 classes in the corpus, the optimal achievable F-measure varied between 0.963 and 0.0. Eleven classes had an F-measure of 0.7 or higher, see Fig. 2. Surprisingly, the best results were achieved with the DOM tree kernel, whereas



**Fig. 2.** Newspaper corpus: Optimal F-measures (cross validation) for the binary classification problems (class vs. rest) using LeftTK

the SetTK and the SoftTK performed worse (not displayed in Fig. 2). Being much too complex, we could not compute any result at all with the StringTK.

## 5 Conclusions

In this paper, we considered four kernels for trees: the LeftTK which is a straightforward extension of the parse tree kernel to XML documents; the set tree kernel, which in contrast to the DOM tree kernel allows the permutation of children; the soft tree kernel, which is an extension of the set tree kernel, that employs a “fuzzy” comparison of node positions and therefore favors trees with more similar orderings of child sequences; and, last, the so-called string-tree kernel, which extends the partial tree kernel by Moschitti [9]. It is possible to prove the correctness of the functions as kernels. We presented a new, straightforward method for including node properties via sub-kernels.

Although the StringTK can be seen as the “nicest” kernel, it seems to render relatively useless when the structures become too big. In this case, the LeftTK might still be a sensible choice in some cases. Generally speaking, the usefulness of a particular kernel depends on the concrete application at hand. In terms of efficiency and accuracy, the soft tree kernel seems to be slightly preferable to the StringTK.

It is obvious that DOM trees can be handled using kernels for general graphs (e.g., [3,15]). Graph kernels do not take advantage of the particular structure of trees and usually have a very high computational complexity. It might still be worth investigating how well they do on trees.

**Acknowledgments.** We thank Alessandro Moschitti (University of Rome, Italy) for helpful discussions, and Alexander Mehler and Olga Pustyl'nikov (University of Bielefeld, Germany) for providing the newspaper and a version of the SUSANNE corpus.

## References

1. Mehler, A., Gleim, R., Dehmer, M.: Towards structure-sensitive hypertext categorization. In: Spiliopoulou, M., Kruse, R., Borgelt, C., Nürnberger, A. (eds.) Proceedings of the 29th Annual Conference of the German Classification Society, Springer, Heidelberg (2005)
2. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
3. Gärtner, T.: A survey of kernels for structured data. SIGKDD Explorations 5(2), 49–58 (2003)
4. Schoelkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge (2002)
5. Collins, M., Duffy, N.: Convolution kernels for natural language. In: NIPS, pp. 625–632 (2001)
6. Moschitti, A.: A study on convolution kernels for shallow statistic parsing. In: ACL, pp. 335–342 (2004)
7. Haussler, D.: Convolution Kernels on Discrete Structure. Technical Report UCSC-CRL-99-10, University of California at Santa Cruz, Santa Cruz, CA, USA (1999)
8. Kashima, H., Koyanagi, T.: Kernels for semi-structured data. In: ICML, pp. 291–298 (2002)
9. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 318–329. Springer, Heidelberg (2006)
10. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.J.C.H.: Text classification using string kernels. Journal of Machine Learning Research 2, 419–444 (2002)
11. Geibel, P., Wysotzki, F.: Learning relational concepts with decision trees. In: Saitta, L. (ed.) Machine Learning: Proceedings of the Thirteenth International Conference, pp. 166–174. Morgan Kaufmann, San Francisco (1996)
12. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines, Software (2001), available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
13. Sampson, G.: English for the Computer: The Susanne Corpus and Analytic Scheme: SUSANNE Corpus and Analytic Scheme. Clarendon Press (1995)
14. Mehler, A., Geibel, P., Pustyl'nikov, O., Herold, S.: Structural classifiers of text types. LDV Forum (to appear, 2007)
15. Geibel, P., Jain, B.J., Wysotzki, F.: Combining recurrent neural networks and support vector machines for structural pattern recognition. Neurocomputing 64, 63–105 (2005)



# Textual Energy of Associative Memories: Performant Applications of Enertex Algorithm in Text Summarization and Topic Segmentation

Silvia Fernández<sup>1,2</sup>, Eric SanJuan<sup>1</sup>, and Juan Manuel Torres-Moreno<sup>1,3,\*</sup>

<sup>1</sup> Laboratoire Informatique d'Avignon, BP 1228 F-84911 Avignon Cedex 9 France

<sup>2</sup> Laboratoire de Physique des Matériaux, CNRS UMR 7556, Nancy, France

<sup>3</sup> École Polytechnique de Montréal - Département de génie informatique  
CP 6079 Succ. Centre Ville H3C 3A7, Montréal (Québec), Canada  
{silvia.fernandez,eric.sanjuan,juan-manuel.torres}@univ-avignon.fr  
<http://www.lia.univ-avignon.fr>

**Abstract.** In this paper we present a Neural Network approach, inspired by statistical physics of magnetic systems, to study fundamental problems of Natural Language Processing (NLP). The algorithm models documents as neural network whose Textual Energy is studied. We obtained good results on the application of this method to automatic summarization and Topic Segmentation.

**Keywords:** Automatic Summarization, Topic Segmentation, Statistical Methods, Statistical Physics.

## 1 Introduction

Hopfield [12] took as a starting point physical systems like the magnetic Ising model (formalism resulting from statistical physics describing a system composed of units with two possible states named spins) to build a Neural Network (NN) with abilities of learning and retrieving of patterns. The capacities and limitations of this Network, called associative memory, were well established in a theoretical frame in several studies [12]: the patterns must be not correlated to obtain free error retrieving, the system saturates quickly and only a little fraction of the patterns can be stored correctly. As soon as their number exceeds  $\approx 0,14N$ , any pattern is recognized. This situation strongly restricts the practical applications of Hopfield Network. However, in NLP, we think that it is possible to exploit this behavior. Vector Space Model (VSM) [3] represents the sentences of a document into vectors. These vectors can be studied as Hopfield NN. With a vocabulary of  $N$  terms, it is possible to represent a sentence as a chain of  $N$  active neurons (words are present) or inactive neurons (words are absent). A document with  $P$  sentences is formed of  $P$  chains in the vector space  $\mathcal{E}$  of dimension  $N$ . These vectors are correlated according to the shared words. If thematics are close, it is reasonable to suppose that the degree of correlation will

---

\* Corresponding author.

be very high. That is a problem if we want to store and retrieve these representations from a Hopfield NN. However, our interest does not relate in retrieving, but on studying the interactions between the terms and the sentences. From these interactions we have defined the Textual Energy of a document. It can be useful, for example, to score or to detect changes between sentences. We have developed a model which makes possible to use the concept of Textual Energy in automatic summarization or topic segmentation tasks. We present in Section 2 a short introduction to the model of Hopfield. In Section 3, we show an extension of this approach in Natural Language Processing. We use elementary notions of graph theory to give an interpretation of Textual Energy like a new measure of similarity. In Section 4 we apply our algorithms to the generation of automatic summaries and the detection of topic boundaries, before concluding and presenting some prospects.

## 2 The Model of Hopfield

Certainly the most important contribution of Hopfield to the theory of NN was the introduction of the notion of energy that comes from the analogy with the magnetic systems. A magnetic system is constituted of a set of  $N$  small magnets called spins. These spins can turn according to several directions. The simplest case is represented by the Ising model which considers only two possible directions: up ( $\uparrow$ , +1 or 1) or down ( $\downarrow$ , -1 or 0). The Ising model is used in several systems which can be described by binary variables [4]. A system of  $N$  binary units has  $\nu = 1, \dots, 2^N$  possible configurations (patterns). In the Hopfield model the spins correspond to the neurons, interacting with the Hebb learning rule [1]:

$$J^{i,j} = \sum_{\mu=1}^P s_{\mu}^i s_{\mu}^j \quad (1)$$

$s^i$  et  $s^j$  are the states of neurons  $i$  and  $j$ . Autocorrelations are not calculated ( $i \neq j$ ). The summation concerns the  $P$  patterns to store. This rule of interaction is local, because  $J^{i,j}$  depends only on the states of the connected units. This model is also known as associative memory. It has the capacity to store and to retrieve certain number of configurations of the system, because the Hebb rule transforms these configurations into attractors (minimal local) of the energy function [1]:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N s^i J^{i,j} s^j \quad (2)$$

Clearly the energy is a function of the system configuration, that is, of the state (of activation or non-activation) of all these units. If we present a pattern  $\nu$ , every spin will undergo a local field  $h^i = \sum_{j=1}^N J^{i,j} s^j$  induced by the others  $N$  spins (figure 1). Spins will align themselves according to  $h^i$  in order to restore

<sup>1</sup> The connections are proportionals to the correlation between neurons' states [2].

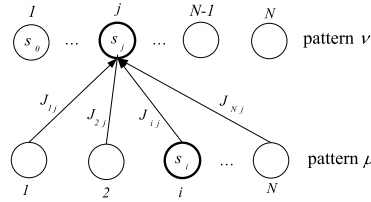


Fig. 1. Field  $h_i$  created by the units of the pattern  $\mu$  affects the pattern  $\nu$

the stored pattern that is the nearest one to the presented pattern  $\nu$ . We will not detail the pattern retrieving method<sup>2</sup>, because our interest will concern the distribution and the properties of the energy of the system (2). This monotonic and decreasing function had only been used to show that the retrieving is convergent. VSM<sup>3</sup> transforms documents in an adequate space where a matrix  $S$  contains the information of the text in the form of bags of words. We can consider  $S$  as the configuration set of a system which we can calculate its energy.

### 3 Applications in NLP

Documents are pre-treated with classical algorithms of functional words filtering<sup>3</sup>, normalization and lemmatisation [6,7] to reduce the dimensionality. A bag of words representation produces a matrix  $S_{[P \times N]}$  of frequencies consisting in  $\mu = 1, \dots, P$  sentences (lines);  $\sigma_\mu = \{s_\mu^1, \dots, s_\mu^i, \dots, s_\mu^N\}$  and a vocabulary of  $i = 1, \dots, N$  terms (columns).

$$S = \begin{pmatrix} s_1^1 & s_1^2 & \dots & s_1^N \\ s_2^1 & s_2^2 & \dots & s_2^N \\ \vdots & \vdots & \ddots & \vdots \\ s_P^1 & s_P^2 & \dots & s_P^N \end{pmatrix}; \quad s_\mu^i = \begin{cases} TF^i & \text{if word } i \text{ exists} \\ 0 & \text{elsewhere} \end{cases} \quad (3)$$

Because the presence of the word  $i$  represents a spin  $s^i \uparrow$  with a magnitude given by its frequency  $TF^i$  (its absence by  $\downarrow$  respectively), a sentence  $\sigma_\mu$  is therefore a chain of  $N$  spins. We differ from [1] on two points:  $S$  is a whole matrix (its elements take absolute frequential values) and we use the elements  $J^{i,i}$  because this autocorrelation makes possible to establish the interaction of the word  $i$  among the  $P$  sentences, which is important in NLP. We apply Hebb's rule (in matricial form) to calculate the interactions between  $N$  terms of the vocabulary:

$$J = S^T \times S \quad (4)$$

Each element  $J^{i,j} \in J_{[N \times N]}$  is equivalent to the calculation of (1). The Textual Energy of interaction between patterns (figure 1) (2) can be expressed:

$$E = -\frac{1}{2} S \times J \times S^T; \quad E_{\mu,\nu} \in E_{[P \times P]} \quad (5)$$

<sup>2</sup> However the interested reader can consult, for example, [1,5,2].

<sup>3</sup> Filtering of numbers and stop-words.

$E_{\mu,\nu}$  represents the energy of interaction between patterns  $\mu$  and  $\nu$ .

### 3.1 Textual Energy: A New Similarity Measure

At this level we are going to explain theoretically the nature of the links between sentences that Textual Energy infers. To do that, we use some elementary notions of the graph theory. The interpretation that we are going to do, is based on the fact that the matrix (5) can be rewritten:

$$E = -\frac{1}{2}S \times (S^T \times S) \times S^T = -\frac{1}{2}(S \times S^T)^2 \tag{6}$$

Let us consider the sentences as sets  $\sigma$  of words. These sets constitute the vertices of the graph. We draw an edge between two of these vertices  $\sigma_\mu, \sigma_\nu$  every time they share at least a word in common  $\sigma_\mu \cap \sigma_\nu \neq \emptyset$ . We obtain the intersection graph  $I(S)$  of the sentences (see an example of four sentences in figure 2). We evaluate these pairs  $\{\sigma_1, \sigma_2\}$ , which we call edges, by the exact number  $|\sigma_1 \cap \sigma_2|$  of words that share the two connected vertices. Finally, we add to each vertex  $\sigma$  an edge of reflexivity  $\{\sigma\}$  valued by the cardinal  $|\sigma|$  of  $\sigma$ . This valued intersection graph is isomorphic to the adjacency graph  $G(S \times S^T)$  of the square matrix  $S \times S^T$ . In fact,  $G(S \times S^T)$  contains  $P$  vertices. There is an edge between two vertices  $\mu, \nu$  if and only if  $[S \times S^T]_{\mu,\nu} > 0$ . If it is the case, this edge is valued by  $[S \times S^T]_{\mu,\nu}$  and this value corresponds to the number of words in common between the sentences  $\mu$  and  $\nu$ . Each vertex  $\mu$  is balanced by  $[S \times S^T]_{\mu,\mu}$ , which corresponds to the addition of an edge of reflexivity. It results that the matrix of Textual Energy  $E$  is the adjacency matrix of the graph  $G(S \times S^T)^2$  in which:

- the vertices are the same ones that those of the intersection graph  $I(S)$ ;
- there is an edge between two vertices each time that there is a way of length 2 in the intersection graph;
- the value of an edge  $(\sigma_\mu, \sigma_\nu)$ : a) where  $\mu = \nu$  (loop) is the sum of the squares of the values of adjacent edges, and b) the sum of the products of the values of the edges on any path of length 2 between  $\sigma_\mu$  and  $\sigma_\nu$  otherwise. These paths can include loops.

From this representation we deduce that the matrix of Textual Energy connects at the same time sentences having common words because it includes the

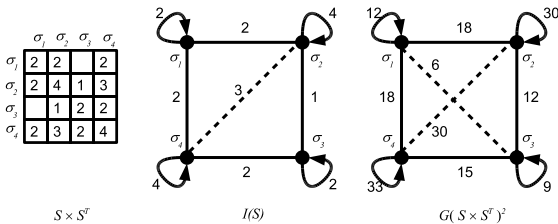


Fig. 2. Adjacency graphs from the matrix of energy

intersection graph, as well as sentences which share the same neighbourhood without necessarily sharing the same vocabulary. So, two sentences  $\sigma_1, \sigma_3$  not sharing any word in common but for which there is at least one third sentence  $\sigma_2$  such that  $\sigma_1 \cap \sigma_2 \neq \emptyset$  and  $\sigma_3 \cap \sigma_2 \neq \emptyset$ , will be connected all the same.

## 4 Experiments and Results

Textual Energy can be used as a similarity measure in NLP applications. In an intuitive way, this similarity can be used in order to score the sentences of a document and thus separate those which are relevant from those which are not. This leads immediately to a strategy for automatic summarization by extraction of sentences. Another approach, less evident, consists in using the information of this energy (seen as a spectrum or numerical signal of the sentence) and to compare with the spectrum of all the others. A statistical test can then indicate if this signal is similar to the signal of other sentences grouped together in segments or not. This can be seen as a detection of thematic boundaries in a document.

### 4.1 Mono-Document Generic Summarization

Under the hypothesis that the energy of a sentence  $\mu$  reflects its weight in the document, we applied (6) to summarization by extraction of sentences [8,9]. The summarization algorithm includes three modules. The first one makes the vectorial transformation of the text using filtering, lemmatisation/stemming and standardization processes. The second module applies the spin model and compute the matrix of textual energy (6). We obtain the weighting of a sentence  $\nu$  by using its absolute energy values, by sorting according to  $\sum_{\mu} |E_{\mu,\nu}|$ . So, the relevant sentences will be selected as having the biggest absolute energy. Finally, the third module generates summaries by displaying and concatenating the relevant sentences. The two first modules are based on the Cortex system<sup>4</sup>. French texts<sup>5</sup> choosed are: *3-melanges* made up of three topics, *Puces* of two topics and *J'accuse* (Emile Zola's letter). Three texts of the Wikipedia in English were analysed, *Lewinsky*, *Quebec* and *Nazca Lines*<sup>6</sup>. We evaluated the summaries produced by our system with ROUGE 1.5.5 [11], which measures the similarity, according to several strategies, between a candidate summary (produced automatically) and summaries of reference (created by humans). In table 11 we compare the performances of the energy method against Mead system<sup>7</sup> that produces only English summaries (symbols  $\emptyset$  in table), Copernic Summarizer<sup>8</sup>, Cortex and a Baseline where the sentences were randomly selected. The compression rate was variable (following the size of the texts) and computed as a

<sup>4</sup> The Cortex system [10] is an unsupervised summarizer of relevant sentences using several metrics controlled by an algorithm of decision.

<sup>5</sup> <http://www.lia.univ-avignon.fr/chercheurs/torres/recherche/cortex>

<sup>6</sup> [http://en.wikipedia.org/wiki/Quebec\\_sovereignty\\_movement; Monica\\_Lewinsky; Nazca\\_lines](http://en.wikipedia.org/wiki/Quebec_sovereignty_movement;Monica_Lewinsky;Nazca_lines)

<sup>7</sup> <http://tangra.si.umich.edu/clair/md/demo.cgi>

<sup>8</sup> <http://www.copernic.com>

**Table 1.** ROUGE-2 (R2) and SU4 score recall. 25%: *3-melanges* (8 ref), *Puces* (8 ref), *Québec* (8 ref) and *Nazca* (6 ref) ; 12%: *J'accuse* (6 ref); 20%: *Lewinsky* (7 ref).

Corpus	Mead		Copernic		Enertex		Cortex		Baseline	
	R2	SU4	R2	SU4	R2	SU4	R2	SU4	R2	SU4
<i>3-melanges</i>	∅	∅	0.4231	0.4348	<i>0.4958</i>	<b>0.5064</b>	<b>0.4968</b>	<b>0.5064</b>	0.3074	0.3294
<i>Puces</i>	∅	∅	<b>0.5775</b>	<b>0.5896</b>	0.5204	0.5336	<i>0.5360</i>	<i>0.5588</i>	0.3053	0.3272
<i>J'accuse</i>	∅	∅	0.2235	0.2707	<i>0.6146</i>	<i>0.6419</i>	<b>0.6316</b>	<b>0.6599</b>	0.2177	0.2615
<i>Lewinsky</i>	0.4756	0.4744	0.5580	0.5610	<i>0.5611</i>	<i>0.5786</i>	<b>0.6183</b>	<b>0.6271</b>	0.2767	0.2925
<i>Quebec</i>	0.4820	0.3891	0.4492	0.4859	<i>0.5095</i>	<i>0.5377</i>	<b>0.5636</b>	<b>0.5872</b>	0.2999	0.3524
<i>Nazca</i>	0.4446	0.4671	0.4270	0.4495	<b>0.6158</b>	<b>0.6257</b>	<i>0.5894</i>	<i>0.5966</i>	0.3041	0.3288

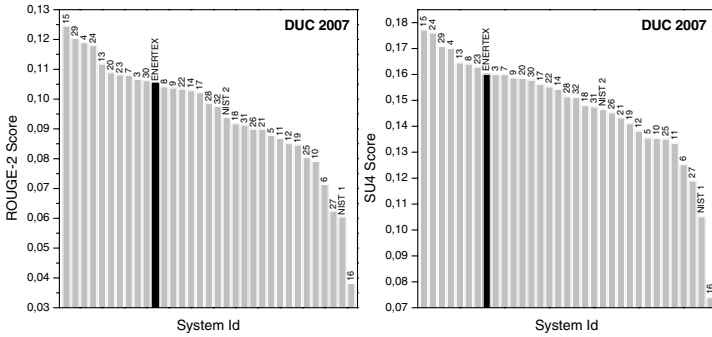
rate on the number of sentences in text. The best performances are in bold and those in 2d position are in italic (all scores). Enertex is a powerful summarizing system (it obtains 3 firsts places and 7 second), close to Cortex system.

## 4.2 Query Oriented Multi-document Summarization

The main task of the NIST-Document Understanding Conference DUC'07<sup>9</sup> is given 45 topics and their 25 document clusters, to generate 250-word fluent summaries that answer the question(s) in the topics statements. In order to calculate the similarity between every topic and the sentences contained in the corresponding cluster we have used Textual Energy (2). Consequently the summary is formed with the sentences that present the maximum interaction energy with the topic. We describe now the process of summary construction using the matricial forms of  $J$  and  $E$  (4 and 5). First, the 25 documents of a cluster are concatenated into a single document following chronological order. Then the Textual Energy between the topic, view as a supplementary sentence, and each of the other sentences in the document is computed using (5). We construct the summary by sorting the most relevant values in the row of matrix  $E$  which correspond to interaction energy of the topic vs. the document.

**Redundancy removal.** In general, in multi-document summarization there is a significant probability of including duplicated information. To avoid this problem, a redundancy elimination strategy has to be implemented. Our system does not include any linguistic processing, then our non-redundancy strategy consists in comparing the energy values of sentences in the generated summary. We suppose that (in a long corpora) the probability that two sentences have the same values of energy is very small. Then we detected the duplicated sentences (with exactly the same energy value) and we replace them by the following ones in the score table. Another strategy, enabling to diversify the content, is to omit long sentences. The threshold with which we obtained the best results was two times the average of the number of words per sentence. Figure 3 shows the

<sup>9</sup> <http://www-nlpir.nist.gov/projects/duc>

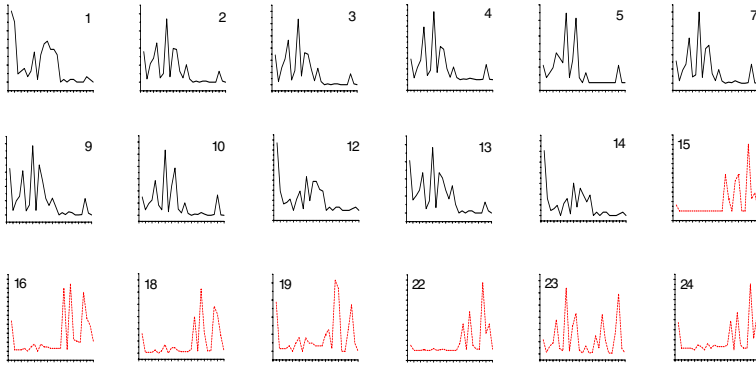


**Fig. 3.** Recall ROUGE-2 and SU4 of the 30 participants in DUC'07 and two baselines

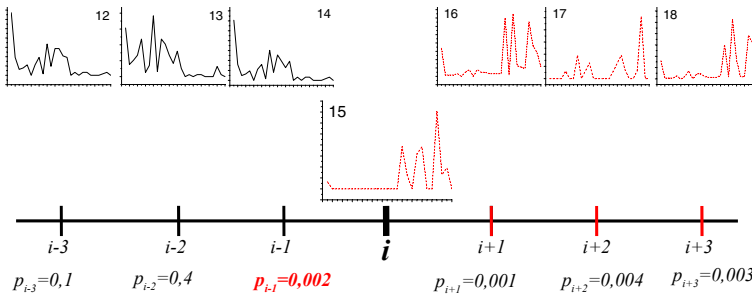
position of our system in the ROUGE automatic evaluation comparing to the 30 participants and two baselines –ID's 1 (random) and 2 (generic summarization system)–.

### 4.3 Topic Segmentation

Several strategies have been developed to segment a text thematically. Most of them are based on Markov models [12], classification of the terms [13,14], lexical chains [15] or on PLSA model [16], that estimates the probabilities of the terms to belong to latent semantic classes. In an original way, we have used the matrix of energy  $E$  (6). This choice makes possible to adapt to new topics and to remain independent from document language. We show in figure 4 the energy of interaction between some sentences of a text made up of two topics. Given that (6) is capable of detecting and of balancing the neighbourhood of a sentence, we can notice a similarity between the curves of the one (bold line) and the other topics (dotted line). In order to compare energies between themselves we have used Kendall's  $\tau$  coefficient of correlation. Given two sentences  $\mu$  and  $\nu$ , we estimate the probability  $P[\mu \neq \nu]$  of being in distinct topics by the probability of  $[\tau(x, y) > \tau(E_{\mu..}, E_{\nu..})]$ . This is done using the normal approximation of Kendall's  $\tau$  law valid if vectors  $E_{\mu..}, E_{\nu..}$  have more than 30 terms.  $\tau$  coefficient does not depend on exact energy values, only on their rank in the vectors  $E_{\mu..}, E_{\nu..}$ . Basically, it evaluates the degree of concordance between two rankings and makes possible robust non parametric statistical test of agreement between two judges classifying a set of  $P$  objects using the fact that  $P[\tau(x, y) > \tau(E_{\mu..}, E_{\nu..})] = 1$  if the ranking vectors associated with  $E_{\mu..}$  and  $E_{\nu..}$  are two statistically independent variables. Here the judges are two sentences that classify all other sentences based on the interaction energy. We shall say that it is almost sure that two sentences  $\mu$  and  $\nu$  are in the same topic if  $P[\mu \neq \nu] > 0.05$ . We have used this test to find the thematic borders between segments. As illustrated in figure 5, a sentence is considered to be at the border of one segment if it is almost sure that: 1/ it is in the same topic as at least two over the three previous sentences; and 2/ it is not in the same topic as at least



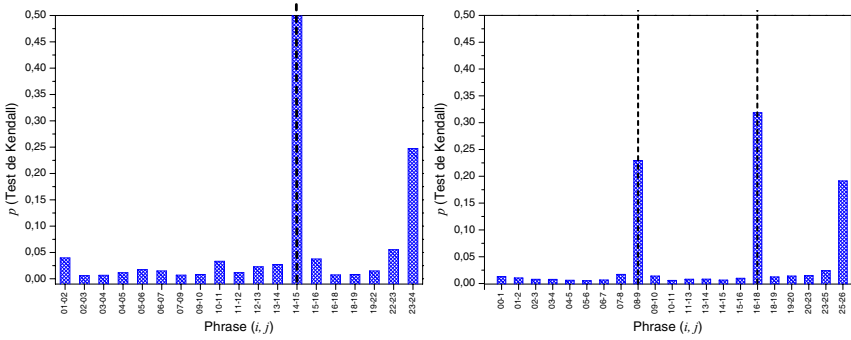
**Fig. 4.** Textual Energy of *2-melanges*. In continuous line, the energy of the sentences of 1<sup>th</sup> topic, in dotted line that of 2<sup>th</sup>. The change of shape of the curves between sentences 14-15 corresponds to a topic boundary. The horizontal axis indicates the number of sentence in the order of the document. The vertical axis, the Textual Energy of the showed sentence vs. others.



**Fig. 5.** Kendall's  $\tau$  in window.  $p_{i\pm k}$  = probability of concordance between  $i \pm k$  and  $i$ .

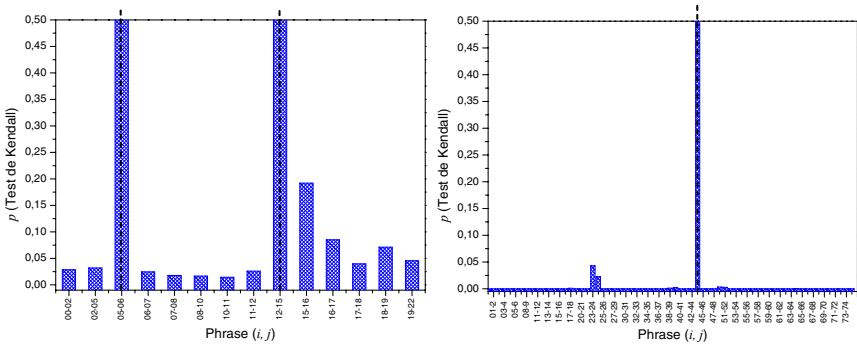
two over the three following sentences. We have implemented this approach of topic segmentation as a slippery window of seven sentences. As the window is moving on, the sentence on its center is compared to all other sentences in the window based on Kendall's  $\tau$  coefficient. If a border is found then the window jumps over the next three sentences. Our programs have been optimized for standard PERL 5 libraries. Figures 6 and 7 show the detection of the boundaries for the texts with 2 and 3 topics. The true boundaries are indicated in dotted line. For the text *3-mélanges*, the test found two borders between the segments 8-9 and 16-18. In both cases, that corresponds indeed to the thematic boundaries. The third (false) boundary was indicated between sentences 14-15 of the text *2-mélanges*. It deserves to be commented on. If we look at figure 4 we can notice that energy of the sentence 23 is very different from that of the sentences 22 or 24. Sentence 23 presents a curve overlapping the two topics. It is the reason why the test cannot identify it like pertaining to the same class. This reasoning





**Fig. 6.** Topic segmentation for the text *2-mélanges* (2 topics, on the left) and *3-mélanges* (3 topics, on the right)

can be extended to all other false borders. We show in figure 7 the boundary detection for texts with 3 and 4 thematics. For the text *physique-climat-chanel* we have detected three boundaries between the sentences 5-6 and 12-15, which corresponds to the real boundaries. For the text in English with two topics the test found one boundary between the segments 44-45 which also corresponds to the real one. In another experiment, we have compared our system to two oth-



**Fig. 7.** Topic segmentation for the text in French with 3 topics *physique-climat-chanel* on the left and in English *Quebec-Lewinsky* on the right

ers: LCseg [17] and LIA\_seg [15] that are based on lexical chains. The corpus of reference was built by [15] from articles of the newspaper *Le Monde*. It is composed of sets of 100 documents where each one corresponds to the average size of the predefined segments. A document is composed of 10 segments (9 borders) extracted from articles of different topics selected at random. The scores are calculated with [18], used in the topic segmentation. This function measures the difference between the real boundaries and those found automatically in a slippery window: the smaller the value is, the more the system is performant.

LIA\_seg depends on a parameter which gives place to various performances (that is why the evaluation of this system gives rise to a range of values). Our method, that uses much less parameters as we do not make any assumption on the number of topics to detect, obtains close performances to the systems in the state of the art. In table 2 we show these results as well as the average number of borders found by Enertex.

**Table 2.** Windiff for LCseg, LIA\_seg and Enertex (variable size segments)

Segment size (sentences)	LCseg	LIA_seg	Enertex (Found boundaries)	
<b>9-11</b>	0.3272	<b>(0.3187-0.4635)</b>	0.4134	7.10/9
<b>3-11</b>	0.3837	<b>(0.3685-0.5105)</b>	0.4264	7.15/9
<b>3-5</b>	0.4344	(0.4204-0.5856)	<b>0.4140</b>	5.08/9

## 5 Conclusion and Perspectives

We have introduced the concept of Textual Energy based on approaches of NN that have enabled us to develop a new algorithm of automatic summarization. Several experiments have shown that our algorithm is adapted to extract relevant sentences. The majority of the topics are approached in the final digest. The summaries are obtained independently of the text size, topics and languages (except for the preprocessing part), and a few quantity of noise is tolerated. Query-guided summaries has been obtained by introducing the topic as supplementary sentence. Some concluding tests on the DUC'07 corpora were realized. We also have studied the problem of topic segmentation of the documents. The method, based on the energy matrix of the system of spins, is coupled with a robust statistical non-parametric test based on Kendall's  $\tau$ . The results are very encouraging. A criticism of this algorithm could be that it requires the handling (produced, transposed) of a matrix of size  $[P \times P]$ . However the graph representation performs these calculations in time  $P \log(P)$  and in space  $P^2$ .

## References

1. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. National Academy of Sciences 9, 2554–2558 (1982)
2. Hertz, J., Krogh, A., Palmer, G.: Introduction to the theorie of Neural Computation. Addison Wesley, Redwood City, CA (1991)
3. Salton, G., McGill, M.: Introduction to modern information retrieval. Computer Science Series. McGraw-Hill, New York (1983)
4. Ma, S.: Statistical Mechanics. World Scientific, Philadelphia, CA (1985)
5. Kosko, B.: Bidirectional associative memories. IEEE Transactions Systems Man, Cybernetics 18, 49–60 (1988)
6. Porter, M.: An algorithm for suffix stripping. Program 14, 130–137 (1980)
7. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge (2000)

8. Mani, I., Maybury, M.T.: Automatic Text Summarization. MIT Press, Cambridge (1999)
9. Radev, D., Winkel, A., Topper, M.: Multi Document Centroid-based Text Summarization. In: ACL 2002 (2002)
10. Torres-Moreno, J.M., Velázquez-Morales, P., Meunier, J.: Condensés de textes par des méthodes numériques. In: JADT. vol. 2, pp. 723–734 (2002)
11. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: WAS 2004 (2004)
12. Amini, M.R., Zaragoza, H., Gallinari, P.: Learning for sequence extraction tasks. In: RIAO 2000 pp. 476–489 (2000)
13. Caillet, M., Pessiot, J.F., Amini, M., Gallinari, P.: Unsupervised learning with term clustering for thematic segmentation of texts. In: RIAO 2004, pp. 648–657 (2004)
14. Chuang, S.L., Chien, L.F.: A practical web-based approach to generating Topic hierarchy for Text segments. In: ACM IKM, Washington, pp. 127–136 (2004)
15. Sitbon, L., Bellot, P.: Segmentation thématique par chaînes lexicales pondérées. In: TALN 2005, vol. 1 pp. 505–510 (2005)
16. Brants, T., Chen, F., Tsochantaridis, I.: Topic-based document segmentation with probabilistic latent semantic analysis. In: CIKM 2002, Virginia, USA, pp. 211–218 (2002)
17. Galley, M., McKeown, K.R., Fosler-Lussier, E., Jing, H.: Discourse segmentation of multi-party conversation. In: ACL-2003, Sapporo, Japan, pp. 562–569 (2003)
18. Pevzner, L., Hearst, M.: A critique and improvement of an evaluation metric for text segmentation. In: Computational Linguistic. vol. 1, pp. 19–36 (2002)

# A New Hybrid Summarizer Based on Vector Space Model, Statistical Physics and Linguistics

Iria da Cunha<sup>1</sup>, Silvia Fernández<sup>2,4</sup>, Patricia Velázquez Morales,  
Jorge Vivaldi<sup>1</sup>, Eric SanJuan<sup>2</sup>, and Juan Manuel Torres-Moreno<sup>2,3,\*</sup>

<sup>1</sup> Institute for Applied Linguistics, Universitat Pompeu Fabra, Barcelona, España  
{[iria.dacunha](mailto:iria.dacunha), [jorge.vivaldi@upf.edu](mailto:jorge.vivaldi@upf.edu)}

<sup>2</sup> Laboratoire Informatique d'Avignon, BP1228, 84911 Avignon Cedex 9, France  
{[silvia.fernandez](mailto:silvia.fernandez), [eric.sanjuan](mailto:eric.sanjuan), [juan-manuel.torres](mailto:juan-manuel.torres)}@univ-avignon.fr

<sup>3</sup> École Polytechnique de Montréal/DGI, Montréal (Québec), Canada

<sup>4</sup> Laboratoire de Physique des Matériaux, CNRS UMR 7556, Nancy, France

**Abstract.** In this article we present a hybrid approach for automatic summarization of Spanish medical texts. There are a lot of systems for automatic summarization using statistics or linguistics, but only a few of them combining both techniques. Our idea is that to reach a good summary we need to use linguistic aspects of texts, but as well we should benefit of the advantages of statistical techniques. We have integrated the Cortex (Vector Space Model) and Enertex (statistical physics) systems coupled with the Yate term extractor, and the Disicosum system (linguistics). We have compared these systems and afterwards we have integrated them in a hybrid approach. Finally, we have applied this hybrid system over a corpora of medical articles and we have evaluated their performances obtaining good results.

## 1 Introduction

Nowadays automatic summarization is a very prominent research topic. This field has been investigated since the sixties, when techniques based on the frequency of terms [19] or on cue phrases [10] were used. Afterwards other techniques, using textual positions [7,18], Bayesian models [16], Maximal Marginal Relevance [12] or discourse structure [22,23,27] were used. In this work, we focus in medical summarization. We do that because, as [1] indicates, nowadays this is a very important area with a very big amount of information that should be processed, so our work aims to help to solve this problem. As well, we are interested in analyzing the techniques used to summarize texts of specialized areas, specifically the scientific-technical ones, so in the future we will extend this work to other domains as chemistry, biochemistry, physics, biology, genomics, etc. We work with the genre of medical papers because this kind of texts are published in journals with their corresponding abstracts written by the authors, and we employ them to compare with the summaries of our systems in order to carry out the final

---

\* Corresponding author.

evaluation. Another motivation to carry out this work is that, although there are a lot of systems for automatic summarization using statistics [6,16,31] or linguistics [2,22,27,33], there are only a few of them combining both criteria [2,3,20,26]. Our idea is that to arrive to a good summary we need to use linguistic aspects of texts, but as well we should benefit of the advantages of statistical techniques. On the basis of this idea, we have developed a hybrid system that takes profit of different aspects of texts in order to arrive at their summaries. To do this, we have integrated three models in this system. Cortex is based in statistics [34], Enertex is based on the Textual Energy [11] and Disicosum is a semiautomatic summarization system that integrates different linguistic aspects of the textual, lexical, discursive, syntactic and communicative structure [9]. In this paper, we have compared these three systems, and afterwards we have integrated them in our hybrid system. Finally, we have applied this system over a corpora of medical articles in Spanish. The resulting summaries have been evaluated with ROUGE [17] obtaining good results. We present a brief experiment in order to observe the influence of the annotator in the tagging process. In Section 2 we describe the three systems that our hybrid system includes. In Section 3 we explain how their integration was carried out. In Section 4 we present the experiments and evaluation, and in Section 5 some conclusions are extracted.

## 2 Systems Used in our Hybrid Approach

### 2.1 Vector Space Models Combining Term Extraction

We tested two different methods for document summarizing, both are based on the Vector Space Model (VSM) [29]. The first method, Cortex, supposes that word frequency can be estimated on the whole set of documents represented as an inverted file. Enertex is inspired in statistical physics, codes a document as a system of spins, and then it computes the "Textual Energy" between sentences to score them. Both systems are coupled with Yate, a term extraction system in order to improve performances in the sentences extraction task.

**Cortex System.** (*Cortex es Otro Resumidor de TEXtos*) [34] is a single-document extract summarization system using an optimal decision algorithm that combines several metrics. These metrics result from processing statistical and informational algorithms on the VSM representation. In order to reduce the complexity, a preprocessing is performed on the topic and the document: words are filtered, lemmatized or/and stemmed. A representation in bag-of-words produces a  $S[P \times N]$  matrix of frequencies/absences of  $\mu = 1, \dots, P$  sentences (rows) and a vocabulary of  $i = 1, \dots, N$  terms (columns). This representation will be used in Enertex system as well. Cortex system can use up to  $\Gamma=11$  metrics [34] to evaluate the sentence's relevance. Some metrics are the angle between the title and each sentence, metrics using the Hamming matrix (matrix where each value represents the number of sentences in which exactly one of the terms  $i$  or  $j$  is present), the sum of Hamming weights of words per segment, the Entropy, the Frequency, the Interactions and others. The system scores sentences with a

decision algorithm combining the normalized metrics. Two averages are calculated, a positive one  $\lambda_s > 0.5$  and a negative one  $\lambda_s < 0.5$  tendencies ( $\lambda_s = 0.5$  is ignored). The decision algorithm combining the vote of each metric is:

$$\sum \alpha = \sum_{\nu} (|\lambda_s^{\nu}| - 0.5); |\lambda_s^{\nu}| > 0.5; \sum \beta = \sum_{\nu} (0.5 - |\lambda_s^{\nu}|); |\lambda_s^{\nu}| < 0.5 \quad (1)$$

The attributed value to every sentence  $s$  is calculated in the following way:

$$\begin{aligned} \text{if } (\sum \alpha > \sum \beta) \text{ then Score}_s &= 0.5 + \frac{\sum \alpha}{\Gamma} : \text{retain } s \\ \text{else Score}_s &= 0.5 - \frac{\sum \beta}{\Gamma} : \text{not retain } s \end{aligned}$$

$\Gamma$  is the number of metrics and  $\nu$  is the index of the metrics.

**Enertex System.** [11] is a Neural Network (NN) approach, inspired by statistical physics, to study fundamental problems in Natural Language Processing, like automatic summarization and topic segmentation. The algorithm models documents as a Neural Network whose Textual Energy is studied. The principal idea is that a document can be treated as a set of interacted units (the words) where each unit is affected by the field created by the others. The associative memory of Hopfield [15] is based on physical systems like the magnetic model of Ising (formalism of statistical physics describing a system with two states units named spins) to build a NN able to store/recovery patterns. Learning is following by Hebb’s rule [14]:

$$J_{i,j} = \sum_{\text{sentences}} s_i s_j \quad (2)$$

and the recovery by the minimisation of the energy of Ising model [14]:

$$E^{\mu,\nu} = \sum s_i^{\mu} J_{i,j} s_j^{\nu} \quad (3)$$

The main limitation of Hopfield NN is its storage capacity: patterns must be not-correlated to obtain free error recovery. This situation strongly restricts its applications, however Enertex exploits this behaviour. VSM represents document sentences into vectors. These vectors can be studied as a NN. Sentences are

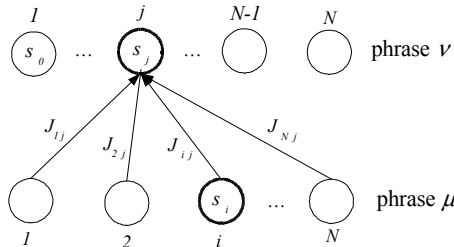


Fig. 1. Field created by terms of the phrase  $\mu$  affects the  $N$  terms of the phrase  $\nu$

represented as a chain of  $N$  active (present term) or inactive (absent term) neurons with a vocabulary of  $N$  terms per document (Fig. 1). A document of  $P$  sentences is formed by  $P$  chains in a  $N$  dimension vector space. These vectors are correlated, according to the shared words. If topics are close, it is reasonable to suppose that the degree of correlation will be high. We compute the interaction between terms by using (2) and the Textual Energy between phrases by (3). Weights of sentences are obtained using their absolute values of energy. The summary consists of the relevant sentences having the biggest values.

**Yate System.** The terms extracted by this tool represent "concepts" belonging to the domain found in the text and their termhood will modify the weights in the term-segment matrix. Yate [35] is a term candidate extraction tool whose main characteristics are: a) it uses a combination of several term extraction techniques and b) it uses EWN<sup>1</sup>, a general purpose lexico-semantic ontology as a primary resource. Yate was designed to obtain all the terms (from the following set of syntactically filtered candidates: <noun>, <noun-adjective> and <noun-preposition-noun>) found in Spanish specialised texts within the medical domain. Yate (Fig. 2) is a hybrid tool that combines the results obtained by a set of term candidate analysers: a) domain coefficient: it uses the EWN ontology to sort the term candidates<sup>2</sup>, b) context: it evaluates each candidate using other candidates present in its sentence context, c) classic forms: it tries to decompose the lexical units in their formants, taking into account the formal characteristics of many terms in the domain and d) collocational method: it evaluates multiword candidates according to their mutual information. The results obtained by this set of heterogeneous methods are combined to obtain a single list of sorted term candidates [35].

## 2.2 Linguistic Model: Disicosum System

The conception of this summarization model of medical articles was done under the hypothesis that professionals of specialized domains (specifically, the medical domain) employ concrete techniques to summarize their texts [9]. [9] have studied a corpora containing medical articles and their abstracts in order to find which kind of information should be selected for a specialized summary and, afterwards, to do generalizations to be included in their model of summarization. Another starting point of this model was the idea of that different types

<sup>1</sup> EWN (www.illc.uva.nl/EuroWordNet) is a multilingual extension of WordNet (wordnet.princeton.edu), a lexico-semantic ontology. The basic semantic unit in both resources is the "synset" that groups together several single/multi words that can be considered synonyms in some contexts. Synsets are linked by means of semantic labels. Due to polysemy, a single lexical entry can be attached to several synsets.

<sup>2</sup> This module locates zones in EWN with high density of terms. The precision obtained when performing in isolation depends on many factors, such as the degree of polysemy of the term candidate or the relative density of terms of the EWN zone. The coverage is high with the obvious limitation of being all or part of the components of the term candidate in EWN. See [36] for details.

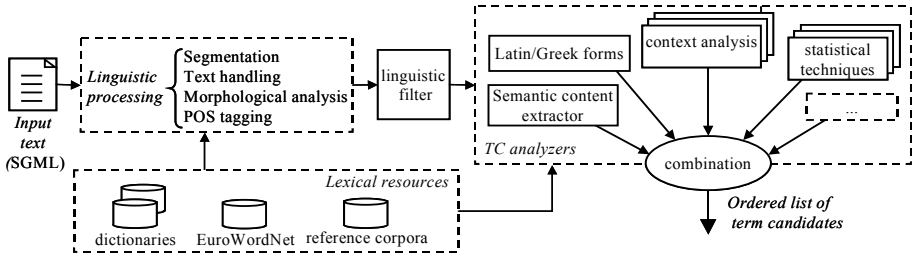


Fig. 2. Architecture of Yate

of linguistic criteria should be used to have a good representation of texts, and in this way exploit the advantages of each criteria. This idea is quite new because, generally, automatic summarization systems based on linguistics use one type of criteria (as we have mentioned above, terms in [19]; textual position in [7,18]; discursive structure in [22,33], etc.), but not the combination of different linguistic criteria. [9] have found linguistic clues that come from the textual, lexical, discursive, syntactic and communicative structures. The system is formed by rules concerning each of those five structures. In the first place, the textual rules of the system indicate that: i) The summary should contain information from each section of the article: Introduction, Patients and methods, Results and Conclusions [32]. ii) Sentences in the following positions should be given an extra weight: the 3 first sentences of the Introduction section, the 2 first sentences of the Patients and methods and the Results sections, and the 3 first and the 3 last sentences of the Conclusions section. In the second place, the system contains lexical rules of two types: a) Lexical rules increasing the score to sentences containing: i/ Words of the main title (except stop words), ii/ Verbal forms in 1st plural person, iii/ Words of a list containing verbs (*to analyse, to observe, etc.*) and nouns (*aim, objective, summary, conclusion, etc.*) that could be relevant, iv) Any numerical information in the Patients and method and the Results sections. b) Lexical rules eliminating sentences containing: i/ References to tables/figures (linguistic patterns show that only a part of sentence should be eliminated: *As it is shown in Table... In Figure 4 we can observe...*), ii/ References to statistical/computational aspects: *computational, program, algorithm, coefficient, etc.*, iii/ References to previous work: *et al* and some linguistic patterns, for example, "determinant + noun (work|study|research|author)". Exceptions: *this study, our research...* iv/ References to definitions: *it is/they are defined by/as...*

Finally, the system includes discursive rules and rules combining discursive structure with syntactic and communicative structure. In order to formalize these rules we follow two theoretical frames: the Rhetorical Structure Theory (RST) [21] and the Meaning-Text Theory (MTT) [24,25]. The RST is a theory of the organization of the text that characterizes its structure as a hierarchical tree containing discursive relations (Elaboration, Concession, Condition, Contrast, Union, Evidence, etc.) between its different elements, that are called



nucleus and satellites. The MTT is a theory that integrates several aspects of the language. In our work, on the one hand, we use its conception of the deep syntax of dependencies, that represents a sentence as a tree where lexical units are the nodes and the relations between them are marked as Actants and the Attributive, Appenditive and Coordinative relations. On the other hand, we use the distinction between Theme and Rheme, that is part of the communicative structure of the MTT. Some examples of the these rules are<sup>3</sup>:

- IF S is satellite<sub>CONDITION</sub> C THEN KEEP S  
[If these patients require a superior flow,] S [it is probably that it is not well tolerated.] N
- IF S is satellite<sub>BACKGROUND</sub> B THEN ELIMINATE S  
~~[Persons who don't want eat and with a complex of fatness have anorexia.]~~ S [We have studied the appearance of complications in anorexic patients.] N
- IF S is satellite<sub>ELABORATION</sub> E1 AND S elaborates on the Theme of the nucleus of E1 THEN ELIMINATE S  
[Persons who don't want eat and with a complex of fatness have anorexia.] N ~~[One of the problems of these patients is the lack of self esteem.]~~ S
- IF S is satellite<sub>ELABORATION</sub> E1 AND S is ATTR THEN ELIMINATE S  
[They selected 274 controls,] N ~~[that hypothetically would have had the same risk factors.]~~ S

### 2.3 Limitations and Solutions for the Model's Implementation

For the implementation of the textual and lexical rules of the model there are no problems because there are preanalysis tools: a segmentation tool developed at the Institute for Applied Linguistics and the Spanish TreeTagger [30]. But we found some problems for the full implementation of the model. The first one, is that there are no parsers able to obtain the discursive structure in Spanish texts. There is one [22,23] for English, and a current project for the Portuguese [28]. The second one, is that there is not known parser to obtain the communicative structure in any language. There are only a few publications about it, as for example [13]. The third one, is that, although there are some syntactic parsers of dependencies for Spanish [5,4], their results, at the moment, are not so reliable as the system needs. So, the solution was to simulate the output of these parsers. Thus, a semiautomatic discursive and syntactic-communicative XML tagger was designed, in order to tag texts and afterwards apply the linguistic summarization model over them [8]. To tag these texts, an annotation interface, where the user can choose the relation between the different elements (nucleus and satellites) of the texts, has been developed. It has to be taken into account that this tagging can be done in two stages. First, the user can detect the relations between sentences and, afterwards, he may find more relations inside each sentence (if any). The final result will be a representation of the text in form of a relations tree, over which the summarization system will be applied. Figure 3 shows the architecture of Disicosum. It is necessary to point out that the rules of the model are applied over the text of each section separately.

<sup>3</sup> N=nucleus, S=satellite. Text underlined will be eliminated.

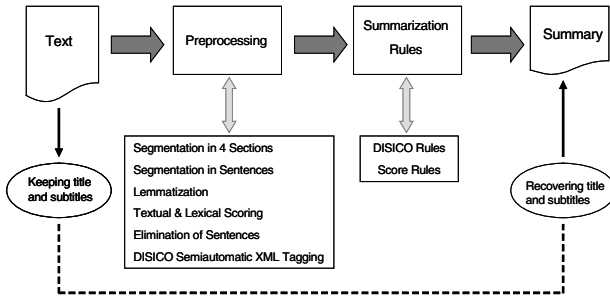


Fig. 3. Architecture of the medical summarization model

### 3 Our Hybrid Approach of Automatic Text Summarization

We have already presented the main characteristics of the different components that our hybrid system integrates. This section briefly will show how such components are integrated in a single hybrid summarization system. Figure 4 presents the system architecture. Firstly, the system applies the elimination rules presented in Section 2.2 over the original text, which produces a reduction  $\approx 20\text{--}30\%$  in its length. Over this reduced text, the system applies separately the Cortex, Enertex and Disicosum systems. A Decision Algorithm processes the normalized output of systems as follow: in the first step, the algorithm chooses the sentences selected by the three systems. If there are no consensus, it chooses the sentences selected by two systems. Finally, if there are sentences selected by only one system, the algorithm gives priority to the sentences with the biggest score.

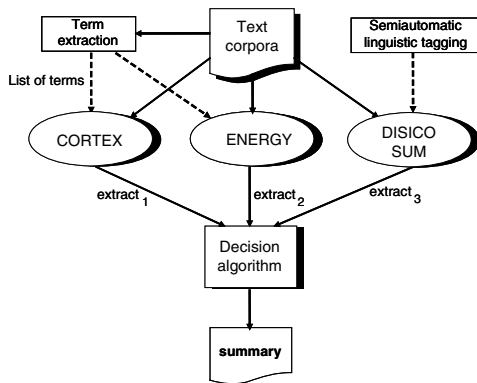


Fig. 4. Architecture of the hybrid summarization system

## 4 Experiments and Evaluation

The corpora used for testing contains 10 Spanish medical papers of the *Medicina Clínica* journal. Before applying the system, texts were semiautomatically tagged with the developed interface by 5 people (2 texts each one). Afterwards, we have compared the summaries produced by the 3 systems, giving to them as input the articles reduced by applying the elimination rules (c.f. 2.2). Also we have created 2 baselines in order to include them in the performance comparison. The difference between them was that Baseline<sub>1</sub> was made from the original article, and Baseline<sub>2</sub> was made from the original article reduced by the application of the elimination rules mentioned above (2.2). To evaluate the summaries we have compared them with the abstracts written by the authors, using ROUGE [17]. In order to interpret the results, it has to be taken into account that authors' summaries are abstracts, while the summaries of our system are extracts. For the application of ROUGE, we have used a Spanish lemmatization and a stop-word list. To set the length of the summaries, we have computed the average number of sentences in each section, present in the author's summaries. Then, we decided to include in our summaries one additional sentence per section. This decision was made because we have noticed that usually authors give, for example, one sentence with different contents in their abstracts, but in their articles they give those contents in separate sentences. In short, it was an empirical decision in order to not lose information. Finally, the system chooses 2 sentences for Introduction and Discussion sections, 3 sentences for the Patients and methods section, and 4 sentences for Results section (11 sentences altogether). In order to analyze the performance of the hybrid system that we present in this article, we have applied it over the ten articles of our corpora, obtaining their summaries. The evaluation results are shown in Table 1. We have used ROUGE measures despite the fact that only one reference abstract is provided. Nevertheless, ROUGE measures provide a standard way of comparing abstracts based on bi-grams and guarantees the reproducibility of our experiments. To evaluate the impact of the quality of semi-automatic tagging on Disicosum performance, two

**Table 1.** Comparison the ROUGE values between different summaries

System	ROUGE-2		SU4	
	Median 1	Median 2	Median 1	Median 2
Hybrid system	<b><u>0.3638</u></b>	<b><u>0.3808</u></b>	<b><u>0.3613</u></b>	<b><u>0.3732</u></b>
Disicosum	<b><u>0.3572</u></b>	<b><u>0.3956</u></b>	<b><u>0.3359</u></b>	<b><u>0.3423</u></b>
Cortex	<b><u>0.3324</u></b>	0.3324	<b><u>0.3307</u></b>	<b><u>0.3255</u></b>
Enertex	0.3105	0.3258	0.3155	0.3225
Cortex on full text	0.3218	<b><u>0.3329</u></b>	0.3169	0.3241
Enertex on full text	<b><u>0.3598</u></b>	<b><u>0.3598</u></b>	<b><u>0.3457</u></b>	<b><u>0.3302</u></b>
Baseline <sub>1</sub>	0.2539	0.2688	0.2489	0.2489
Baseline <sub>2</sub>	0.2813	0.3246	0.2718	0.3034

documents among the ten were tagged in a restricted time (30 min per text) and the others without time restrictions. Therefore, the coherence of the linguistic tagging on these texts is expected to be better than for the two texts tagged in restricted time. Table 1 gives the median score of each system on the ten documents (column median1) and on the reduced set of documents tagged without time restrictions (column median2). Font scores depend on the quartile (big fonts for higher quartiles, smaller ones for others). Regarding the median score on the documents which tagging has been done in an unrestricted time, Disicosum abstracts seem to be the closest to author abstract according to ROUGE-2 measure. According to SU-4 Disicosum is the best among the individual systems but the hybrid system has a better score. Regarding the whole set of texts, the median score of Disicosum is lower than the previous one, however it remains among the higher ones for individual systems. This shows that the quality of the linguistic model tagging has a direct impact on the summary quality meanwhile the tagging is carried out independently from the summarisation purpose. Cortex and Enertex systems have been tested directly on full texts or after segmenting texts into independent sections. The segmentation preprocess is part of Disicosum. The second best individual system according to these results seems to be Enertex on full text. It appears that Enertex works better without the indication that the summary should contain elements coming from each section of the text. An explanation could be that Enertex compares all sentences two by two. The more sentences the text has, the better is the vector representation of the sentence in the system. Cortex uses more local criteria since it has been built to efficiently summarise large corpora. On short texts, the lack of frequent words reduces the efficiency of the system however it appears here that it can take into account the structural properties of the texts. Looking at the hybrid system, the experiment shows that it improves the proximity with the author's abstract in all cases except for ROUGE-2 when considering human linguistic tagging done without time restriction. Finally, we have carried out another experiment: we use the same text (number 6) annotated by five different people and their summaries generated by Disicosum as reference models, and we have decided to compute ROUGE tests over all other systems. The idea is to find which system is closer to models. Results and an example of summary are shown in Table 2. Cortex and Enertex are the closer systems to the linguistic model. In other words, performance of Disicosum and the two numerical summarizers used are equivalents.

**Table 2.** ROUGE values for five different summaries models and example of summary

	ROUGE-2	SU4	<i>Evaluación de las vías de acceso venoso innecesarias en un servicio de urgencias. Fundamento. Los accesos venosos son uno de los proce dimientos que con más frecuencia se practican en los servicios de urgencias con un fin terapéutico, que puede ser inmediato o no, en función de la sintomatología que presente el paciente o el diagnóstico de sospecha inicial. El objetivo del presente trabajo fue evaluar el volumen de pacientes a quienes se les practica un acceso venoso, estimar cuántos de ellos son innecesarios y el coste económico que ello genera.</i>
Author	0.1415	0.1710	
Cortex	<b>0,7281</b>	<b>0.7038</b>	
Enertex	<b>0,7281</b>	<b>0.7038</b>	
Baseline <sub>1</sub>	0.3059	0.2920	
Baseline <sub>2</sub>	0.3662	0.3740	

## 5 Conclusions

We show in this paper, on the one hand, that the summaries produced by statistical methods (Cortex and Enertex) are similar to the summaries produced by linguistic methods. On the other hand, we have proved that combining statistics and linguistics in order to develop a hybrid system for automatic summarization gives good results, even better than those ones obtained by each method (statistical or linguistic) separately. Finally, we have tested that the Disicosum system offer very similar summaries although different annotators tag the original text (that is, the annotators give different discourse trees). Other tests, comparing several summaries produced by doctors and our hybrid system, may be realized. Extensions to other domains and languages will also be considered.

## References

1. Afantenos, S.D., Karkaletsis, V., Stamatopoulos, P.: Summarization of medical documents: A survey. *Artificial Intelligence in Medicine* 2(33), 157–177 (2005)
2. Alonso, L., Fuentes, M.: Integrating cohesion and coherence for Automatic Summarization. In: *EACL 2003 Student Session, ACL, Budapest*, pp. 1–8 (2003)
3. Aretoulaki, M.: COSY-MATS: A Hybrid Connectionist-Symbolic Approach To The Pragmatic Analysis Of Texts For Their Automatic Smmarization. PhD thesis, University of Manchester, Institute of Science and Technology, Manchester (1996)
4. Asterias, J., Comelles, E., Mayor, A.: TXALA un analizador libre de dependencias para el castellano. *Procesamiento del Lenguaje Natural* 35, 455–456 (2005)
5. Attardi, G.: Experiments with a Multilanguage Non-Projective Dependency Parser. In: *Tenth Conference on Natural Language Learning, New York* (2006)
6. Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: *Intelligent Scalable Text Summarization Workshop, ACL, Madrid, Spain* (1997)
7. Brandow, R., Mitze, K., Rau, L.: Automatic condensation of electronic publications by sentence selection. *Inf. Proc. and Management* 31, 675–685 (1995)
8. da Cunha, I., Ferraro, G., Cabre, T.: Propuesta de etiquetaje discursivo y sintáctico-comunicativo orientado a la evaluación de un modelo lingüístico de resumen automático. In: *Conf. Asoc. Española de Lingüística Aplicada, Murcia* (2007)
9. da Cunha, I., Wanner, L.: Towards the Automatic Summarization of Medical Articles in Spanish: Integration of textual, lexical, discursive and syntactic criteria. In: *Crossing Barriers in Text Summarization Research, RANLP, Borovets* (2005)
10. Edmundson, H.P.: New Methods in Automatic Extraction. *Journal of the Association for Computing Machinery* 16, 264–285 (1969)
11. Fernández, S., SanJuan, E., Torres-Moreno, J.M.: Énergie textuelle de mémoires associatives. *Traitement Automatique des Langues Naturelles*, pp.25–34 (2007)
12. Goldstein, J., Carbonell, J., Kantrowitz, M., Mittal, V.: Summarizing text documents: sentence selection and evaluation metrics. In: *22nd Int. ACM SIGIR Research and development in information retrieval*, pp. 121–128. ACM Press, New York (1999)
13. Hajicova, E., Skoumalova, H., Sgall, P.: An Automatic Procedure for Topic-Focus Identification. *Computational Linguistics* 21(1) (1995)
14. Hertz, J., Krogh, A., Palmer, G.: *Introduction to the theorie of Neural Computation*. Addison-Wesley, Redwood City (1991)

15. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. *National Academy of Sciences* 9, 2554–2558 (1982)
16. Kupiec, J., Pedersen, J.O., Chen, F.: A trainable document summarizer. In: *SIGIR-1995*, New York, pp. 68–73 (1995)
17. Lin, C.Y.: Rouge: A Package for Automatic Evaluation of Summaries. In: *Workshop on Text Summarization Branches Out (WAS 2004)*, pp. 25–26 (2004)
18. Lin, C., Hovy, E.: Identifying Topics by Position. In: *ACL Applied Natural Language Processing Conference*, Washington, pp. 283–290 (1997)
19. Luhn, H.P.: The automatic creation of Literature abstracts. *IBM Journal of research and development* 2(2) (1959)
20. A., M.: Towards a Hybrid Abstract Generation System. In: *Int. Conf. on New Methods in Language Processing*, Manchester, pp. 220–227 (1994)
21. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. *Text* 8(3), 243–281 (1988)
22. Marcu, D.: The rhetorical parsing, summarization, and generation of natural language texts. PhD thesis, Dep. of Computer Science, University of Toronto (1998)
23. Marcu, D.: *The Theory and Practice of Discourse Parsing Summarization*. Institute of Technology, Massachusetts (2000)
24. Mel'cuk, I.: *Dependency Syntax: Theory and Practice*. Albany: State University Press of New York (1988)
25. Mel'cuk, I.: *Communicative Organization in Natural Language. The semantic-communicative structure of sentences*. John Benjamins, Amsterdam (2001)
26. Nomoto, T., Nitta, Y.: A Grammatico-Statistical Approach to Discourse Partitioning. In: *15th Int. Conf. on Comp. Linguistics*, Kyoto, pp. 1145–1150 (1994)
27. Ono, K., Sumita, K., Miike, S.: Abstract generation based on rhetorical structure extraction. In: *15th Int. Conf. on Comp. Linguistics*, Kyoto, pp. 344–348 (1994)
28. Pardo, T., Nunes, M., Rino, M.: DiZer: An Automatic Discourse Analyzer for Brazilian Portuguese. In: *Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI)*, vol. 3171, pp. 224–234. Springer, Heidelberg (2004)
29. Salton, G., McGill, M.: *Introduction to modern information retrieval*. Computer Science Series. McGraw Hill Publishing Company, New York (1983)
30. Schmid, H.: Probabilistic Part-of-speech Tagging Using Decision Trees. In: *International Conference on New Methods in Language Processing* (1994)
31. Silber, H.G., McCoy, K.F.: Efficient text summarization using lexical chains. In: *Intelligent User Interfaces*, pp. 252–255 (2000)
32. Swales, J.: *Genre Analysis: English in Academic and Research Settings*. Cambridge University Press, Cambridge (1990)
33. Teufel, S., Moens, M.: Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics* 28 (2002)
34. Torres-Moreno, J.M., Velázquez-Morales, P., Meunier, J.G.: Condensés de textes par des méthodes numériques. In: *JADT*, St. Malo, pp. 723–734 (2002)
35. Vivaldi, J.: *Extracción de candidatos a término mediante combinación de estrategias heterogéneas*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, 2001.
36. Vivaldi, J.: Medical term extraction using the EWN ontology. In: *Terminology and Knowledge Engineering*, Nancy, pp. 137–142 (2002)

# Graph Decomposition Approaches for Terminology Graphs

Mohamed Didi Biha<sup>1</sup>, Bangaly Kaba<sup>2</sup>, Marie-Jean Meurs<sup>3</sup>,  
and Eric SanJuan<sup>3,\*</sup>

<sup>1</sup> LANLG, 33 rue Louis Pasteur, 84000 Avignon, France  
mohamed.didi-biha@univ-avignon.fr

<sup>2</sup> LIMOS, Université Blaise Pascal Clermont 2, 63177 AUBIERE cedex, France  
kaba@isima.fr

<sup>3</sup> LIA, Université d'Avignon, BP 1228 84911 Avignon, Cedex 9, France  
{marie-jean.meurs,eric.sanjuan}@univ-avignon.fr

**Abstract.** We propose a graph-based decomposition methodology of a network of document features represented by a terminology graph. The graph is automatically extracted from raw data based on Natural Language Processing techniques implemented in the TermWatch system. These graphs are Small Worlds. Based on clique minimal separators and the associated graph of atoms: a subgraph without clique separator, we show that the terminology graph can be divided into a central kernel which is a single atom and a periphery made of small atoms. Moreover, the central kernel can be separated based on small optimal minimal separators.

**Keywords:** graph algorithms, graph decomposition, polyhedral approach, text mining, topic visualisation.

## 1 Introduction

Terminology graphs that include explicitly defined properties and relationships developed for human-curated semantic networks, such as controlled ontologies, are used for organizing and communicating information. At the core of these terminologies are discrete elements of knowledge, or entities, which carry meaning. The way in which these entities are arranged and encoded in electronic format is a key concern in informatics [1].

The TermWatch system [2] aims to automatically extract a terminology graph from texts based on Natural Language Processing (NLP) approaches originally introduced in [3].

In this paper we show how these graphs can be structured in coherent sub-networks in order to allow its visualisation and to approximate a real concept network. For that we use two recent graph decomposition approaches. The first one is based on the concept of graph of atoms (an atom is a subgraph without clique separator). The important point is that this decomposition is unique.

---

\* Corresponding author.

However, there is no upper limit to atom size and we have observed that terminology graphs are made of small atoms gravitating at the peripheral of a huge central one. It is necessary to break this central atom into equal parts without losing its internal structure. We show that this can be efficiently accomplished using optimal separators.

The rest of the paper is organised as follows. In section 2 we recall the features of terminology graphs extracted using TermWatch system. In section 3, we formally define the process of graph decomposition into atoms. In section 4 we show how optimal separators can be found. In section 5 we experiment the whole process on a real corpus. Finally, we conclude on related work and perspectives.

## 2 TermWatch System

This system comprises three modules: a term extractor, a relation identifier which yields the terminological network and a visualisation module.

### 2.1 Term Extraction

This module performs term extraction based on shallow NLP, using the LTPOS tagger and LTChunker<sup>1</sup>. LTChunker identifies simplex noun phrases (NPs), i.e., NPs without prepositional attachments. In order to extract more complex terms, we wrote contextual rules to identify complex terminological NPs, i.e. those with a prepositional attachment. The number of words in a term is not limited. This choice is based on the observation that most concepts in the technical domain are long multi-word terms.

### 2.2 Identifying Semantic Nearest Neighbours (*S*-NN) of Terms

This module identifies the different semantic variants of the same term based on surface and internal linguistic operations between MWTs and the use of an external resource, here WordNet.

Morphological variants are identified using the LTPOS tagger. Lexical variants are identified based on word changes in terms. However, the definition of lexical variants is restricted in order to allow the change of only one word in the same position so as to avoid generating spurious relations. The change can take place either in a modifier position (*T-cell line / fibroblast line*) or in the head position (*T-cell line / T-cell lymphoma*). The head in a noun phrase is the term focus (subject) while the modifier plays the role of a qualifier. Syntactic variants involve structural changes in terms, for instance a permutation: “*retrieval of information*” and “*information retrieval*”. Other syntactic operations called expansions, involve the addition of modifier or head words in a term. Modifier expansions are subdivided into “Insertions” and “Left-expansions”. Head expansions are either expansions solely on the head position of a term or double

---

<sup>1</sup> (C) Andrei Mikheev 1996–2000 (C) LTG, University of Edinburgh 1996–2000.



expansions (both in the modifier and head position). Semantic variants are used to identify more semantically bound terms amongst lexical substitutions as the latter can be noisy, especially on binary terms. For instance, a chain of lexical head substitutions can link all of these terms: *T-cell line*, *T-cell lineage*, *T-cell lymphoma*, *T-cell lysate*, *T-cell malignancy*, *T-cell maturation*, *T-cell mitogen*, *T-cell mitogenesis*. This can capture semantically close terms like “*T cell line*” and “*T cell lineage*” but this would be purely accidental. To identify semantic substitutions amongst the lexical ones, WordNet is used to filter those variants where the substituted words are in a WordNet relation<sup>2</sup>. We distinguish WordNet substitutions according to the two grammatical functions of the substituted word: head or modifier.

The variations described above can be further refined according to the number and position of inserted words for expansion variants. Thus we distinguish further between strong and weak expansions. Strong expansions are those variants where only one word is added (*B cell lymphoma line / human B cell lymphoma line*) while those involving the addition of more than one word are considered as weak expansion variants (*TSH receptor / TSH receptor (TSHR)-specific T cell line*).

### 2.3 Graph Visualisation Module

For visualization purposes, graphs are clustered. For this task, we use a variant of the single link clustering (SLC), called CPCL (Classification by Preferential Clustered Link) originally introduced in [3] to form clusters of keywords related by geodesic paths made of strongest associations. The advantages of SLC clustering are that it produces a unique output and that it runs in linear time on the number of edges. The CPCL variant also has these properties. It merges iteratively clusters of keywords related by an association strongest than any other in the external neighborhood. In other words, CPCL works on local maximal edges instead of absolute maximal values like in standard SLC. CPCL output is unique such as in SLC while reducing the chain effect. We refer the reader to [4] for a detailed description in the graph formalism. The CPCL algorithm has been optimised to run in  $O(|E|)$  time, where  $|E|$  is the number of edges of the graph.

Finally, using the interactive interface AiSee (<http://www.aisee.com>) and its optimized bi-scale force directed layout, we obtain a two level access to the network of terms and clusters.

AiSee needs as input a file in Graph Description Language (GDL). Our GDL generator uses edge width to visualize the strength of the link. Clusters are then represented by ovals whose size depends on the number of clustered vertices. Finally, special clusters can be unfolded in a wrapped form that allows to visualize the transitions to other clusters.

<sup>2</sup> Despite the fact that general resources cannot capture the explicit conceptual relation between specialized domain terms, we still highly improved the precision of the substitutions variants using WordNet, in the sense that 97% of the WordNet substitutions linked semantically related terms.

### 3 Atom Graph Decomposition

#### 3.1 Introduction to Graph Decomposition

We will recall some preliminary graph notions which will be helpful to follow our approach.

A graph is denoted  $G = (V, E)$  where  $V$  is a finite set of vertices and  $E$  is a finite set of edges. The graphs on which we work are undirected.  $G(A)$  is the subgraph induced by a vertex set  $A$  (included in  $V$ ). A clique in a graph is a set of pairwise adjacent vertices. A connected component graph is a maximal vertex set which induces a maximal connected subgraph. A tree is a connected graph without cycle. A graph is said to be chordal graph iff there is no chordless cycle of length more than 3. In the simple graphs that are the trees, the articulations are the vertices which are not leaves (vertices which have at maximum one incident edge). The removal of an articulation vertex defines several subgraphs. To decompose a tree, we copy an articulation in each subgraph its removal defines. Graphs subjacent to corpus are not trees. To decompose them, we use instead of articulation vertex in the case of tree, the groups of vertices called ‘minimal separators’. A subset  $S$  of vertices is a *minimal separator* of a connected graph  $G = (V, E)$  if  $G(V - S)$  has at least two connected components. In general, a graph has an exponential number of minimal separators. However, it has been proved that the number of clique minimal separators (separators that are completely connected) is weak and less than the number of vertices. In fact the decomposition of the graph and the enumeration of all clique minimal separators can be done in linear time  $O(|V||E|)$ . This has been dealt by Tarjan in [5]. This process is based on minimal triangulation algorithms that embed a graph into a chordal graph by the addition of an inclusion-minimal set of edges.

Thus we propose an algorithmic process which decomposes a graph of terms subjacent to a given corpus of textual data into connected groups of terms which are called ‘atoms’ and that do not have clique separators. One of the interesting advantages of this decomposition is that atoms we define are not disjoint, but can have an overlap. The process of decomposition consists in copying a ‘clique minimal separator’ into different parts of the graph, so that each overlap between two ‘atoms’ is a ‘clique minimal separator’. Previous works have proved that the intersection graphs of subtrees in trees are exactly the chordal graphs [6]. In our application, we have found that our graphs of atoms are chordal. To keep the structure of the graph, they have to be copied in each different part defined by their removal. One of the important features of this method is that the decomposition is unique.

#### 3.2 Algorithmic Decomposition

We have introduced our method in [4]. To implement our program, first we computed a minimal triangulation in linear time  $O(n.m)$  followed by the process of graph decomposition dealt by Tarjan. Thus, we produced all clique minimal

separators of the graph. Then we decompose the graph in a set of atoms. If there are cycles with more than three vertices in the graph, they will be retrieved into atoms. The algorithm decomposition is described as below:

**Decomposition algorithm**

**Input** : A graph  $G = (V, E)$  and moplex ordering  $\kappa$ .

**Output** : Set of graph components

**Initialization:**  $i \leftarrow 1, j \leftarrow 1$ .

**begin**

```

    For  $i=1$  to  $|\kappa|-1$  do ;
     $S_i \leftarrow N_G(\kappa(i))$ ;  $N_G(\kappa(i))$  is the neighborhood of  $i$ 
    if  $S_i$  is a clique then
         $C \leftarrow i$  and its neighborhood;
         $C \leftarrow G-C$ ;
         $Comp(j) \leftarrow C \cup S_i$ ;
         $j \leftarrow j + 1$ ;
     $Comp(j) \leftarrow G$  (Last component);

```

**end**

**3.3 Atom Decomposition of Small World Graphs**

A graph is said to be SWG when it simultaneously shows both low diameter and high clustering measure, (i.e., high density of edges in the neighborhood of each vertex). According to [7], the path length  $L(p)$  and the clustering coefficient  $C(p)$  are the two structural measurements that characterize the SWGs.

The usual approach to visualize a SWG consists in computing a decomposition into highly connected components and to offer to the user an abstract view of the network to start with [8].

We adopt a similar approach except that we compute overlapping atoms [4] instead of disjoint connected components. The atoms of a graph can be defined based on the concept of  $(a, b)$ -clique separators.

By definition an atom  $A$  of a graph  $G$  contains at least one complete separator  $S$  of  $G$ , however  $S$  is not a separator of  $A$ . Atoms overlap if they contain the same separator of  $G$ .

In our experiments, we have observed that graphs have a central atom with long cycles that involves almost 50% of the vertices and numerous peripheral atoms of small size that are almost chordal (cycles have less than three elements).

To visualize small atoms and their interactions on a map, we shall define a valued graph exclusively based on the structure of  $G = (V, E)$ . Each atom  $A$  is labeled by the vertex  $w_1$  having the highest degree defined as the number of edges linking  $w_1$  to another vertex  $w_2$  in  $A$ . Atoms having the same label are merged together. The valued graph of atoms that we shall denote by  $G(At) = (V_{At}, E_{At}, a_{At})$  is defined as follows.

The vertex of  $G(At)$  are pairs of the form  $(k, l)$  where  $k$  is a vertex of  $G_k$  and  $l$  is the label of an atom containing  $k$ . An edge  $e = (w_1, w_2)$  is defined between two vertices  $w_1 = (k_1, l_1)$  and  $w_2 = (k_2, l_2)$  if one of the following happens:

1.  $l_1 = l_2$  and  $(k_1, k_2)$  is an edge of  $G$ . In this case the value  $s_{At}$  of the edge  $e$  is set to 1.
2.  $k_1 = k_2$  and there exists a clique separator  $S$  in  $G$  that separates the atom  $l_1$  from the atom  $l_2$ . In this case  $a_{At}(w_1, w_2)$  is set to the ratio between the number of elements in  $S$  and the total number of elements in atoms  $l_1$  and  $l_2$ .

The first case corresponds to edges in atoms. To ensure that the related vertices will not be separated by any clustering procedure, we set the value of such edges to 1, the maximum. The second case deals with edges relating copies of  $G$  vertices in different atoms. This valued graph can be displayed as described here below.

Now to visualize the central atom we shall look for optimal minimal separators that allow to split the atom into parts of equal size.

## 4 Graph Decomposition by Optimal Separators

Combinatorial optimization is a lively field of applied mathematics, combining techniques from combinatorics, linear programming, and the theory of algorithms, to solve optimization problems over discrete structure. Combinatorial optimization searches for an optimum object in a finite collection of objects. Typically, the collection has a concise representation, like a graph, while the number of objects is huge. Combinatorial optimization problems are usually relatively easy to formulate mathematically, but most of them are computationally hard. The basic idea behind polyhedral techniques is to derive a good linear formulation of the set solutions by identifying linear inequalities that can proved to be necessary in the description of the convex hull of feasible solutions.

### 4.1 Polyhedral Approach for Ab-separator Problem

Finding a balanced minimum-weight separator in a  $n$ -vertex graph that partitions the graph into two components of similar sizes, smaller than  $2n/3$ , is relevant in many problems. Formally, the vertex separator problem (VSP) can be stated as follows. The instance consists of a connected undirected graph  $G = (V, E)$ , with  $|V| = n$ , an integer  $\beta(n)$  such that  $1 \leq \beta(n) \leq n$  and a cost  $c_i$  associated with each vertex  $i \in V$ . The problem is to find a partition  $\{A, B, C\}$  of  $V$  such that :

$$E \text{ contains no edge } (i, j) \text{ with } i \in A, j \in B, \tag{1}$$

$$\max\{|A|, |B|\} \leq \beta(n), \tag{2}$$

$$\sum_{j \in C} c_j \text{ is minimized} \tag{3}$$

The vertex separator problem (VSP) is NP-hard [9]. In 2005, Egon Balas and Cid De Souza provide the first polyhedral study of the vertex separator problem (VSP) [10]. Recently, Didi Biha and Meurs [in preparation], starting from the Balas and De Souza's work, studied the vertex separator polyhedron and gave several new valid inequalities for this polyhedron.

### 4.2 The Polyhedron of Separators

For a given graph  $G = (V, E)$ , we consider the particular case of (VSP) in which two non-adjacent vertices  $a$  and  $b$  are given and we look for a partition  $\{A, B, C\}$  which satisfies (1) and (2) with  $a \in A, b \in B$  and  $|C|$  is minimum. This particular case is called *ab-separator* problem in this paper. We can solve (VSP) by solving at most  $\frac{n(n-2)}{2}$  ab-separator problems.

Given the non-adjacent vertices  $a$  and  $b$ , the incidence vector of a partition  $\{A, B, C\}$  of  $V$  which satisfies (1) and (2) with  $a \in A$  and  $b \in B$  is  $X = (x_{1a}, \dots, x_{(n-2)a}, x_{1b}, \dots, x_{(n-2)b}) \in \{0, 1\}^{2(n-2)}$ , with  $x_{ia} = 1 \Leftrightarrow i \in A, x_{ib} = 1 \Leftrightarrow i \in B, \forall i \in V \setminus \{a, b\}$ .

Let  $P_{ab}$  be the polyhedron associated to the ab-separator, i.e.  $P_{ab} = Conv \{X \in R^{2(n-2)} : X \text{ is an incidence vector for some ab-separator partition } \{A, B, C\}\}$ .

Let  $\Gamma_{ab}$  be a simple chain between  $a$  and  $b$ . Let  $I(\Gamma_{ab})$  be the set of intern vertices of  $\Gamma_{ab}$ . The inequality  $\sum_{i \in I(\Gamma_{ab})} (x_{ia} + x_{ib}) \leq |I(\Gamma_{ab})| - 1$  is valid for  $P_{ab}$  (4.2).

In fact, if  $\{A, B, C\}$  is an ab-separator partition, then every chain from  $a$  to  $b$  contains at least one vertex of  $C$ . For all couple of non-adjacent vertices  $(i, j) \in V$ , let  $\alpha_{ij}$  be the maximum number of disjoint chains between  $i$  and  $j$ .

### 4.3 Model

Our model is formally described as follows:

Data :

- A connected undirected graph  $G = (V, E)$ , with  $|V| = n$ ,
- An integer  $\beta(n)$
- $a \in A, b \in B$  virtual vertices,
- $\alpha_{min} = \text{Min}\{\alpha_{ij}, i \in V, j \in V, (i, j) \notin E\}$

The (VSP) can be formulated as the following mixed integer programming :

$$\begin{aligned} \text{Maximize } & \sum_{i=1}^n (x_{ia} + x_{ib}) \\ \text{s.c. :} & \\ & x_{ia} \in \{0, 1\}, \quad \forall i \in V \quad (4) \\ & x_{ia} + x_{ib} \leq 1, \quad \forall i \in V \quad (5) \end{aligned}$$

$$x_{ia} + x_{jb} \leq 1, \quad \forall (i, j) \in E \tag{6}$$

$$x_{ja} + x_{ib} \leq 1, \quad \forall (i, j) \in E \tag{7}$$

$$\sum_{i=1}^n (x_{ia} + x_{ib}) \leq n - \alpha_{min} \tag{8}$$

$$1 \leq \sum_{i=1}^n x_{ia} \leq \lfloor \frac{n - \alpha_{min}}{2} \rfloor \tag{9}$$

$$1 \leq \sum_{i=1}^n x_{ib} \leq \beta(n), \quad 1 \leq \sum_{i=1}^n x_{ia} \leq \beta(n), \tag{10}$$

The constraint (5) is valid since  $A$  and  $B$  are disjoint sets. The constraints (6) and (7) are valid since there is no edge between the two sets  $A$  et  $B$ . The constraint (8) comes from (4.2). Without loss of generality, we may assume that  $|A| \leq |B|$ . Furthermore,  $|A| + |B| \leq n - \alpha_{min}$ , thus  $|A| \leq \lfloor \frac{n - \alpha_{min}}{2} \rfloor$ , that is corresponding to the constraint (9) in our model. The constraints (10) are valid since the ab-separators satisfy (2).

## 5 Application

Based on graph decomposition technique into atoms described here above, the methodology used here for viewing results merges two graphs into the same visual output: the graph of term-term associations and of variation links and, a second graph of author-term association (ATCA here below), enabling us to link authors to the clusters of terms used (research topics) in their publications.

### 5.1 Methodology Summary

The method works in seven phases described hereafter:

1. **Term extraction and selection:** noun phrases (NPs) are extracted from ISI abstracts. NPs are merged together based on COMP relations described in section 2: spelling variants, left expansions, insertions, modifier substitution and WordNet synonyms. The resulting clusters are the connected components of the graph where vertices are terms and there is one edge between each term and its variants. We shall refer to these clusters as Term components. Only Term components with at least two vertices are considered. Each of these Term components are labelled by the NP having the most number of variants. Supplementary NPs are considered based on CLASS relations. These relations are: left expansions and head substitutions on NPs with at least three words. NPs involved in such variations are added as supplementary Term components but not clustered. Each of them is isolated in a separated component. This a way to select NPs based on surface linguistic variation.

2. **Terminology graph extraction:** associations are computed between Term components or authors, this results in the extraction of a new graph where vertices can be both previous term components or authors. An edge is drawn between two of these vertices whenever a valid association is found. Associations are computed in the following way. For each component or author  $x$ , we denote by  $D(x)$  the set of documents where abstracts have at least one NP in  $x$  if  $x$  is a component or where  $x$  is one of the authors otherwise. Then two vertices  $c_1$  and  $c_2$  are associated if there are at least two documents in  $D(c_1).D(c_2)$ . In this case, an equivalence  $E(c_1, c_2)$  coefficient is computed between  $c_1$  and  $c_2$  in the following way:

$$E(c_1, c_2) = \frac{|D(c_1).D(c_2)|^2}{|D(c_1)|.|D(c_2)|}$$

Only associations for which  $E(c_1, c_2) > 0.05$  are considered. Let us call this graph ATCA (Associations between Term Components and Authors).

3. **Graph components:** connected components of ATCA are computed. It is usual that in this kind of graph, there are lots of small components and only one really big component that contains more than two thirds of the vertices. We shall refer to it as the main ATCA component.
4. **Atom decomposition:** the main ATCA component is decomposed into atoms. An atom is a sub-graph where there is no clique separator. Again in this kind of graph, it often occurs that there is a central huge atom and several small ones. We shall refer to the biggest atom as the central ATCA atom.
5. **Peripheral atom layout:** peripheral atoms and their interactions are visualized by generating the atom graph as described in section 3.3. Since atoms overlap, each atom is labelled by its central vertex (the vertex with the highest degree). A vertex in several atoms is duplicated. Each copy is labelled by the vertex label and the atom label. An edge is drawn between two copies of ATCA vertices if they share the same atom label or if they are copies in separated atoms that involve a common ATCA separator.
6. **Central atom separation:** we also implemented the splitting of the central atom based on optimal separators as described in §4.
7. **Central atom visualisation:** the central atom components are visualized based on Single Link Clustering that groups together vertices whose association equivalence coefficient between is higher than any other one in the neighbourhood (local maximums). This allows us to reduce the size while preserving the graph structure.

## 5.2 Results

We experimented this method on the same corpus as [11] on terrorism extracted from ISI bibliographic database. In this corpus, 57,855 NPs were extracted from 3,366 ISI abstracts. These NPs were clustered into 3,293 Term components with at least two NPs. The maximal size of Term components is 30. 8,357 supplementary terms having at least one CLASS variant are added to the set of Term

components. The ATCA graph has 16,258 edges. Its main component has 9,324 edges over 1,070 vertices. This component involve 489 atoms. The central atom has 2,070 edges over 307 vertices and can be splitted in three parts using two separators of four vertices each. All the other atoms have less than 29 vertices.

Upon closer inspection, these three sub-networks corroborate the findings in Chen's study (2006) on the same corpus but on a shorter period (1990-2003). In [11], three major groups of clusters were identified by author co-citation and term networks using CiteSpace II: a cluster on 'body injuries in terrorist bombing', a second bigger cluster on 'health care response to the threat of biological and chemical weapons', a third biggest and more recent cluster on psychological and psychiatric impacts of the september 11, 2001 terrorist attack with terms like 'United States' and 'posttraumatic stress disorder' (PTSD) being very prominent. We found a similar demarcation in the internal structure of the central atom that has the following separators:

1. 'health care provider', 'specific clinical', 'AU: Tonat\_K' and 'physical injury'.
2. 'public health', 'AU: Tracy\_M', 'terrorist attack victim' and 'AU: Pfefferbaum\_B'.

Figure 1 shows the atom graph of the terminology graph computed on this corpus and its central atom before applying the decomposition based on optimal minimal separators.

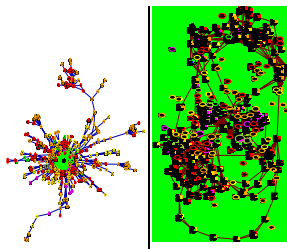


Fig. 1. Atom graph (left) and its central atom (righth)

## 6 Conclusion

To the best of our knowledge, this paper is the first attempt to apply graph atom decomposition to knowledge domain mapping [14,15,16]. The advantage of atom graph decomposition is that it is unique since it is based on the intrinsic structure of the graph. Its main drawback is that small atoms do not always exists in a graph. The results obtained on the corpus used in this experiment tend to show that: atom decomposition is tractable on a large corpus of documents and that central atoms can be separated using optimal separators.

Previous experimentations on other bibliographic corpora dealing with *Information Retrieval*, *genomics* or *Organic Chemistry* have confirmed these results.



## References

1. ME., B., SB., J.: Graph theoretic modeling of large-scale semantic networks. *J. Biomed Inform.* 39(4), 451–464 (2006)
2. SanJuan, E., Ibekwe-SanJuan, F.: Text mining without document context. *Information Processing and Management* 42, 1532–1552 (2006)
3. Ibekwe-SanJuan, F.: A linguistic and mathematical method for mapping thematic trends from texts. In: *Proc. of the 13th European Conference on Artificial Intelligence (ECAI)*, Brighton, UK, pp. 170–174 (1998)
4. Berry, A., Kaba, B., Nadif, M., SanJuan, E., Sigayret, A.: Classification et désarticulation de graphes de termes. In: *Proc. of the 7th International conference on Textual Data Statistical Analysis (JADT 2004)*, Louvain-la-Neuve, Belgium, pp. 160–170 (2004)
5. Tarjan, R.E.: Amortized computational complexity 6(2), 306–318 (1985)
6. Gavril, F.: The intersection graphs of subtrees in trees are exactly the chordal graphs 16, 47–56 (1974)
7. Ferrer-i-Cancho, R., Sole, R.V.: The small world of human language. *Proceedings of The Royal Society of London. Series B, Biological Sciences* 268(1482), 2261–2265 (2001)
8. Auber, D., Chiricota, Y., Jourdan, F., Melancon, G.: Multiscale visualization of small world networks. In: *IEEE Symposium on Information Visualisation*, pp. 75–81. IEEE Computer Society Press, Los Alamitos (2003)
9. Bui, T., Fukuyama, J., Jones, C.: The planar vertex separator problem: Complexity and algorithms. *Manuscript* (1994)
10. Balas, E., de Souza, C.C.: The vertex separator problem: a polyhedral investigation. *Math. Program.* 103(3), 583–608 (2005)
11. Chen, C.: Citespace ii: Detecting and visualizing emerging trends and transient patterns in scientific literature. *JASIST* 57(3), 359–377 (2006)
12. Neumann, A., Gräber, W., Tergan, S.O.: Paris - visualizing ideas and information in a resource-based learning scenario. In: *Knowledge and Information Visualization*, pp. 256–281 (2005)
13. Ganter, B., Wille, R.: *Formal Concept Analysis*. In: *Mathematical Foundations*, Springer, Heidelberg (1999)
14. Braam, R., Moed, H., A., A.V.R.: Mapping science by combined co-citation and word analysis. 2. dynamical aspects. *Journal of the American Society for Information Science* 42(2), 252–266 (1991)
15. Small, H.: Visualizing science by citation mapping. *JASIS* 50(9), 799–813 (1999)
16. Schiffrin, R., Börner, K.: Mapping knowledge domains. In: Schiffrin, R., Börner, K. (eds.) *Publication of the National Academy of Science (PNAS)*, vol. 101(suppl 1), pp. 5183–5185 (2004)

# An Improved Fast Algorithm of Frequent String Extracting with no Thesaurus

Yumeng Zhang<sup>1,2</sup> and Chuanhan Liu<sup>1</sup>

<sup>1</sup>Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200030, China

<sup>2</sup>School of Business, Ningbo University, Ningbo, 315211, China  
zhyumeng@yahoo.com.cn, uuchliu@163.com

**Abstract.** Unlisted word identification is the hotspot in the research of Chinese information processing. String frequency statistics is a simple and effective method of extraction unlisted word. Existing algorithm cannot meet the requirement of high speed in vast text processing system. According to strategies of string length increasing and level-wise scanning, this paper presents a fast algorithm of extracting frequent strings and improves string frequency statistical method. The approach does not need thesaurus, and does not need to word segmentation, but according to the average mutual information to identify whether each frequent string is a word. Compared with previous approaches, experiments show that the algorithm gains advantages such as high speed, high accuracy of 91% and above.

## 1 Introduction

Word segmentation and word extraction are always the hotspot and nodus in the research field of Chinese information processing. Domestic researchers had provided numerous solutions that can be broadly categorized as two kinds, one dictionary-based method, the other non-dictionary-based method. The first one needs the ideal condition of perfect dictionary equipped. However, even if the dictionary is the largest in the world, it cannot collect all the natural language vocabularies [1]. Those unlisted words are some important elements that influence the accuracy of Chinese word segmentation and word extraction. In order to identify automatically these unlisted words, the scholars had provided the method based on rules or statistics to achieve the unlisted word extraction [2-4]. The second method is the main focus in such field. For example, on the base of segmentation, Tan carried through the frequency statistic and gains a great number of new candidate words [5]. Nie used the method of X<sup>2</sup> statistic and general likelihood ratio to calculate inter-word correlation, the unlisted words were obtained automatically from corpus to segment words or construct automatically dictionary [6]. In [7], a word or a phrase is identified from the interval correlation of the string.

Recent research shows that string frequency statistics is a simple and effective approach of identifying the unlisted word [8-11]. In Chinese, a sentence is a string that is made up of the characters one by one and there's no obvious separator in the

text. The more frequently a continuous string occurs, the higher possibility that it forms a word [12]. The process of using the string frequency to extract the unlisted word can be implemented in three steps: (1) using various separation marks to pre-cut the text and splitting it into several phrases; (2) searching for the high-frequency strings from the pre-cut phrase set; (3) processing statistical weighting on the frequent strings and judging whether the frequent strings are meaningful words. Step (2) is the most time-consuming procedure. It determines the overall performance of word extraction. Reference [3] and [4] are basically the same. The latter improved on the algorithm in the former and made the computational time reduced. To increase further the calculating speed, a spatial memory leaps matching algorithm was suggested in reference [13]. Although the computational time of the algorithm had been improved significantly, some high-frequency strings were missed. The quality of unlisted word extraction is unsatisfactory. Besides the long computational time, the string frequency statistics is difficult to filter out those non-word strings with high co-occurrence frequency [14], such as “you duo” (有多), “ke wei”(可为).

Based on the previous researches, a fast algorithm of high-frequency string extraction based on the level-wise scanning is proposed in this paper. Compared with the previous approaches, it has significantly reduced the computational time and improved the quality of word extraction. This method utilizes the ideas of *Apriori*'s association rules mining in Data Mining and uses the property of *Apriori* as much as possible to reduce the search space, so that the efficiency of high frequency string extraction is improved greatly. It provides a statistical way of valid support count, which can be used to filter out frequent but meaningless string and improve the accuracy of word extraction.

## 2 Frequent String Extraction

### 2.1 Definition and Theorem

**Definition 1.** Short sentence, a continuous character string divided by separation marks after the text is pre-processed.

**Definition 2.** String  $x_i = \{w_1 w_2 \cdots w_n\}$ , a string made up of some continuous characters in a short sentence, where  $w_i (1 \leq i \leq n)$  denotes a character,  $n$  denotes the length of the string.

**Definition 3.** Support count  $\sigma(x_i)$ , the times that a string  $x_i$  occurs in the short sentences set.

**Definition 4.** Support degree  $S(x_i)$ , the frequency that a string  $x_i$  occurs in the short sentence set.  $S(x_i) = \sigma(x_i)/A \times 100\%$ , where  $A$  denotes the sum of candidate strings.

**Definition 5.** Frequent string, the string whose occurrence frequency in the short sentences set is greater than or equal to the minimum support threshold (min\_sup).

**Definition 6.** Sub-string, if a string  $q$  contains a string  $p$ , then  $p$  is the sub-string of  $q$ , or  $p$  is the subset of  $q$ .

**Definition 7.** Super-string, if a string  $q$  contains a string  $p$ , then  $q$  is the super-string of  $p$ , or  $q$  is the superset of  $p$ .

**Definition 8.**  $k$ -string ( $k \geq 1$ ), the string that contains  $k$  characters.

*Apriori* is the first association rules mining algorithm that makes the support-based pruning used to control systematically the exponential growth of candidate itemsets [15]. This algorithm uses the prior knowledge of frequent itemsets properties, employs an iterative approach known as a level-wise search, where  $k$ -itemsets are used to explore  $(k+1)$ -itemsets. *Apriori* has an important property that all nonempty subsets of frequent itemsets must also be frequent. This property can be used to reduce the search space. It can substantially improve the efficiency of the level-wise generation of frequent itemsets. The property is also available for fast frequent strings extraction. To make some changes, it can deduce Theorem 1 with the similar property of *Apriori*.

**Theorem 1.** If a sub-string is infrequent, then its super-string is also infrequent. (i.e., all the nonempty sub-strings of frequent strings must be frequent)

**Proof:** Suppose a string  $x_1$  is the sub-string of a string  $x_2$ . That is  $x_1 \subset x_2 \Rightarrow x_2 = x_1 \cup x'$ ; if  $x_1$  is infrequent, then  $x_1$  does not satisfy  $\min\_sup, S(x_1) = \sigma(x_1)/N \leq \min\_sup$ .  $x_2$  is the super-string of  $x_1$ ; it is a string that has added  $x'$  on the base of  $x_1$ . It cannot occur more frequent than  $x_1$ , thus,  $\sigma(x_2) = \sigma(x_1 \cup x') \leq \sigma(x_1) \Rightarrow S(x_2) = \sigma(x_2)/N \leq S(x_1) \leq \min\_sup$ . Thus,  $x_2$  is also infrequent. ■

**Theorem 2.** The shorter the sub-strings cut from strings are, the less the sum of candidate strings will be.

**Proof:** Suppose a string  $X$  whose length is  $N$ , is cut into  $m$  sub-strings  $\{x_1, x_2, \dots, x_m\}$ , the length of sub-string  $x_i$  ( $1 \leq i \leq m$ ) is  $l_i$ , and  $\sum_{i=1}^m l_i = N$ . Let the number of candidate  $k$ -strings produced by  $X$  be  $s_1$ . The sum of candidate  $k$ -string produced by the sub-string set  $\{x_1, x_2, \dots, x_m\}$  is  $s_2$ . Suppose  $l_i \geq k$ .

$$s_1 = N - k + 1$$

$$s_2 = \sum_{i=1}^m (l_i - k + 1) = \sum_{i=1}^m l_i - m(k - 1) = N - k + 1 - (m - 1)(k - 1) \tag{1}$$

thus

$$\frac{s_2}{s_1} = \frac{N - k + 1 - (m - 1)(k - 1)}{N - k + 1} = 1 - \frac{(m - 1)(k - 1)}{N - k + 1} \tag{2}$$

From formula (2), it can be seen that,  $s_1 = s_2$  if the string  $x$  is not cut ( $m=1$ ),  $s_2 \leq s_1$  if the string  $X$  is cut ( $m \geq 2$ ), and  $s_2/s_1$  decreases when the number of cut times ( $m$ ) increases. ■

## 2.2 Text Preprocessing

It can be seen from Theorem 2 that in order to enhance the processing speed, the text should be cut into the short strings as short as possible. Though there is no distinct separator between Chinese words, there are two kinds of separators that can be used. They are absolute separator and conditional separator. The absolute separators are punctuation, numbers, letters, etc; the conditional separators are those single Chinese characters with weak formation ability, numerals, or function words, such as “de” (的), “le” (了), “yi mian” (以免), “bing qie” (并且). They can hardly constitute words with other Chinese characters. Thus, the text can be cut into short sentences set by using these separators. This set is named as  $S$ .

## 2.3 Frequent String Finding Algorithm

The traditional  $n$ -gram extraction approach will generate the large quantities of candidate strings. For example, it can be known from formula (1), a string of 10 characters can generate 45 candidate strings. To compute the support count of each candidate string is going to be time-consuming work. It cannot satisfy the requirement of high speed in vast information processing system. Based on the level-wise scan, the frequent string finding algorithm given in this paper can reduce greatly the size of candidate strings.

In order to enhance the efficiency of frequent string generation, Theorem 1 can be used to compress the search space. Firstly, the set of frequent 1-string, denoted as  $L_1$ , is found.  $L_1$  can be used to find the set  $L_2$  which denotes the set of frequent 2-strings,  $L_2$  can be used to find  $L_3$ , and so on. This algorithm stops until no more frequent  $k$ -strings can be found. It is made of candidate string forming step and pruning step.

(1) Candidate string forming step: In order to find the frequent string set  $L_k$ , this algorithm extracts a candidate  $k$ -string from the short sentence set  $S$ , and denotes it as the candidates set  $C_K$ . According to the sub-strings in  $L_{k-1}$ ,  $C_K$  collects all possible frequent strings. For example, the string  $x_i = \{w_i w_{i+1} \cdots w_{i+k-2}\}$  is not in  $L_{k-1}$ , thus, according to Theorem 1, its super-string  $y_i = \{w_i w_{i+1} \cdots w_{i+k-1}\} = x_i \cup w_{i+k-1}$  can be identified to be infrequent too.

(2) Pruning step: The candidate set  $C_k$  is the superset of frequent string set  $L_k$ . Its members can be frequent or infrequent, but all of the frequent  $k$ -strings are included in  $C_k$ . A scan of short sentence set  $S$  is running. It determines the count of each candidate

in  $C_k$  so that  $L_k$  can be determined (*i.e.*, all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to  $L_k$ ).

$C_k$ , however, is maybe huge, and this could involve heavy computation. According to Theorem 1, any  $(k-1)$ -string that is not frequent cannot be a sub-string of a frequent  $k$ -string. Thus, if any  $(k-1)$ -substring of candidate  $k$ -string is not in  $L_{k-1}$ , the candidate  $k$ -string cannot be frequent. It is not necessary to scan the whole short sentence set before it is removed from  $C_k$ . Therefore, based on the prior knowledge, the algorithm can reduce greatly the search space and increase the search speed.

**Table 1.** Support count of frequent strings

Strings	和谐	谐社	社会	和谐社	谐社会	和谐社会
Support Count	4	2	3	2	2	2

For example, a string "构建社会主义和谐社会, 社区的和谐, 家庭的和谐, 是和谐社会的重要组成部分。" was divided into a set  $S = \{$ "构建社会主义和谐社会", "社区", "和谐", "家庭", "和谐", "是和谐社会", "重要组成部分"}, where  $|S|$  denotes the sum of all characters in the set, thus  $|S| = 29$ .

(1) In the first iteration of the algorithm, each character is a member of the set of candidate 1-strings  $C_1$ , where  $C_1 = \{w_1, w_2, \dots, w_i\}$ . The algorithm scans simply all the phrases in order to count the times of occurrences of each character.

(2) Suppose that the required minimum support count is 2. The set of frequent 1-strings  $L_1$  can be determined. It consists of the candidate 1-strings satisfying the minimum support count, where  $L_1 = \{\{和\}, \{谐\}, \{社\}, \{会\}\}$ .

(3) In order to find the set of frequent 2-strings  $L_2$ , the algorithm truncates in turn the string  $x_i = \{w_i\}$  which consists one character and generates a candidate set of 2-characters. If  $x_i \in L_1, x_{i+1} \in L_1$ , thus its super-string  $y_i = \{w_i w_{i+1}\}$  is frequent possibly. The algorithm collects it into  $C_2$ . So,  $C_2 = \{\text{社会, 和谐, 谐社}\}$ .

(4) Next, the phrases in  $S$  are scanned and the support count of each candidate string in  $C_2$  is accumulated.

(5) The set of frequent 2-strings  $L_2$  is then determined, consisting of those candidate 2-strings in  $C_2$  having the minimum support count. Therefore,  $L_2 = \{\text{和谐, 谐社, 社会}\}$ .

(6) Repeating step (2) - (5), the frequent 3-strings and 4-strings are generated finally. Table 1 lists these frequent strings that contain more than two characters and satisfy the minimum support count.

### 2.4 Valid Support Count Correction

Though the frequent strings achieved by the existing algorithms are sometimes the meaningless strings, they have the minimum support count, but they are not their independent real frequency, just some sub-strings split by super-string. For example, “xie she”(谐社) is the sub-string of “he xie she”(和谐社) and “xie she hui”(谐社会); “he xie she”(和谐社) and “xie she hui”(谐社会) are the sub-strings of “he xie she hui”(和谐社会).

**Definition 9.** Valid support count, the valid support count of a string equals its occur frequency minus the frequency of its most frequent super-string:

$$Valid(x_i) = Fre(x_i) - \max\{Fre(\text{sup}(x_i))\} \tag{3}$$

For example, “he xie”(和谐) occurs 4 times, its super-string “he xie she”(和谐社) and “xie she hui”(谐社会) occur twice. Therefore, its valid support count is (4-2)=2. Also, the valid support counts of “xie she”(谐社) and “she hui”(社会) are zero and once. Therefore, the algorithm outputs the strings (“he xie”(和谐), “he xie she hui”(和谐社会)) whose valid support counts are greater than the minimum support count, and removes strings “xie she”(谐社), “she hui”(社会). The valid support count of frequent strings is listed in table 2.

**Table 2.** Valid support count of frequent strings

Strings	和谐	谐社	社会	和谐社	谐社会	和谐社会
Valid Support Count	2	0	1	0	0	2

### 2.5 Mutual Information-Based String Filtering

After valid support count correction, there are still some random strings made of frequent characters, such as “zhong xing”(中兴), “ye bu”(也不). These strings occur much frequently, but they cannot be identified only by frequency calculating. They need be filtered further. Mutual information (MI) is a commonly used statistical model. It is a kind of measure of the correlation between two random variables  $X$  and  $Y$ . Mutual information between strings shows the correlation of the two strings. The greater the MI value is, the more possibilities that strings may form words or phrases. The string  $x_i = \{w_1 w_2 \dots w_n\}$  can be divided into a sub-string  $x_{j-} = \{w_1 w_2 \dots w_j\}$  and a sub-string  $x_{j+} = \{w_{j+1} w_{j+2} \dots w_n\}$ , ( $1 \leq j < n$ ), the mutual information formula of  $x_i$  is shown as formula (4).

$$MI(x_i) = \log \frac{P(x_i)}{P(x_{j-})P(x_{j+})} = \log \frac{A \times N(x_i)}{N(x_{j-}) \times N(x_{j+})} \tag{4}$$

where  $P(x_i)$ ,  $P(x_{j-})$  and  $P(x_{j+})$  are the frequency of  $x_i$ ,  $x_{j-}$  and  $x_{j+}$  in the candidate string set respectively,  $N(x_i)$ ,  $N(x_{j-})$  and  $N(x_{j+})$  represent the valid

support count of  $x_i$ ,  $x_{j-}$  and  $x_{j+}$ .  $A$  represents the sum of characters of the candidate string set.

Since the extracting strings include two-string and multi-character string, the average mutual information is used to measure multi-character string.

$$MI(x_i) = \frac{1}{n-1} \sum_{j=1}^{n-1} MI(x_{j-}, x_{j+}) \quad 1 \leq j < n \quad (5)$$

### 3 Experimental Results and Analysis

The experiment is based on the T2050, 1G memory, Windows XP platform. The texts with different lengths are selected randomly to test. The algorithms proposed in this paper and the traditional approach of n-gram are applied to test the frequent string extraction. The experimental results on two aspects are listed as follows: (1) the efficiency of frequent strings extraction, (2) the quantity and quality of frequent strings extraction.

#### 3.1 The Efficiency of Extraction

The candidate  $(k+1)$ -string on the base of frequent  $k$ -string repeatedly and circularly until no candidate strings are appeared. The experimental results are shown in Table 3.

The field *id*, *len* represents the code number and the length of the texts respectively, *sen* represents the quantity of gained short sentences after pre-processing,  $S_1, S_2, \dots, S_{10}$  represents the quantity of candidate 1-string, 2-string, ..., 10-string gained by the algorithm presented in this paper;  $NS_1, NS_2, \dots, NS_{10}$  represent the quantity of candidate string 1-string, 2-string, ..., 10-string gained by the traditional algorithm.  $S_1$  and  $NS_1$  are the quantities of 1-string. The support count of each character has exceeded the set threshold value in the experiment. Those 2-strings constituted by them are all regarded as the candidate strings. Therefore,  $S_2$  equals  $NS_2$ . The candidate 2-tring set has already many strings whose support counts are smaller than the *min\_sup*. Their super-strings (3-string) cannot be collected to the candidate 3-string set. Therefore,  $S_3$  is less than  $NS_3$ . According to Table 3, it can be found that the quantity of candidate strings decrease obviously with the increase of string length so that the statistic and computational work reduce. The comparison of the sum of produced candidate strings is shown in Figure 1, where the abscissa is the text length and the longitudinal coordinate represents the sum of the produced candidate strings.

In the aspect of time complexity, the computational time of this algorithm is spends principally on sorting strings. For example, the time spent on quick sorting of all the strings of frequent  $k$ -strings set is less than  $f(k)\log(f(k))$  where  $f(k)$  is the sum of  $k$ -strings. Therefore, the time complexity of this algorithm is  $O(\sum_{k \geq 2} f(k)\log(f(k)))$ .

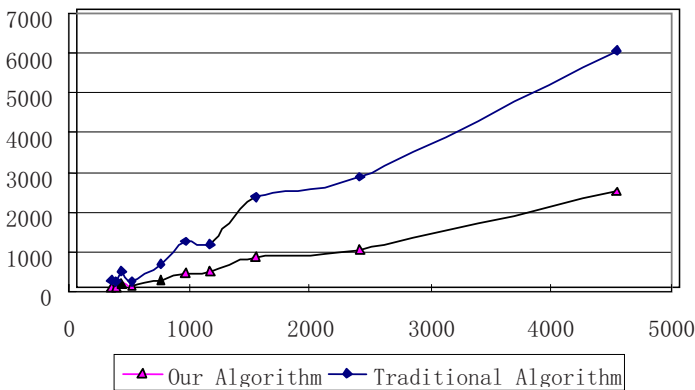
From Table 3, it can be found that the sum of the frequent strings is decreased quickly when their length is more than 3. So, the main computational time is spent on the sorting of the frequent 2-string.



**Table 3.** Quantity comparison of different lengths' candidate string

id	len	Sen	S1	NS1	S2	NS2	S3	NS3	S4	NS4
2	357	48	143	143	93	93	27	76	5	50
5	390	46	171	171	75	75	23	56	7	37
4	448	49	171	171	100	100	52	93	29	81
6	524	68	209	209	101	101	39	76	11	44
8	760	105	270	270	192	192	74	169	17	117
9	969	135	277	277	313	313	118	287	32	218
10	1169	168	337	337	352	352	112	291	28	204
3	1550	211	358	358	519	519	223	511	67	410
7	2414	348	577	577	708	708	217	599	72	459
1	4543	596	639	639	1215	1215	652	1275	267	1049

S5	NS5	S6	NS6	S7	NS7	S8	NS8	S9	NS9	S10	NS10
1	32	0	17	0	8	0	3	0	1	0	0
2	24	1	16	0	10	0	8	0	7	0	6
15	67	6	53	5	40	4	31	3	27	2	24
5	23	3	13	2	6	1	4	0	2	0	1
4	78	0	53	0	34	0	22	0	14	0	9
10	156	3	109	1	77	0	53	0	37	0	26
11	137	3	86	0	52	0	32	0	19	0	10
25	306	10	221	5	164	1	119	0	84	0	60
27	340	15	254	7	189	5	144	4	110	3	80
135	799	90	585	60	426	42	310	34	230	28	169



**Fig. 1.** Quantity comparison of candidate strings

**Table 4.** The effect of frequent word extraction

id	Length of text	Our Algorithm			Traditional Algorithm		
		Number of words	Number of Correct words	Correct Rate	Number of words	Number of Correct words	Correct Rate
2	357	8	8	100%	7	7	100%
5	390	13	12	92.3%	14	12	85.7%
4	448	16	15	93.8%	13	11	84.6%
6	524	16	16	100%	18	16	88.9%
8	760	17	16	94.1%	14	12	85.7%
9	969	17	16	94.1%	19	17	89.5%
10	1169	24	22	91.7%	18	15	83.3%
3	1550	23	22	95.6%	20	18	90.0%
7	2414	39	37	94.9%	34	29	85.3%
1	4543	74	69	93.2%	88	70	79.5%

### 3.2 The Quantity and Quality of Extraction

The sample above is taken as the test object. Our algorithm corrects the final valid support count of frequent string and also calculates the average mutual information of every frequent string by choosing the string whose average mutual information is greater. The experimental results are shown in Table 4.

Compared with the previous algorithms, this paper provides a new algorithm that improved frequent string extraction in both aspects of quantity and quality.

## 4 Conclusions

In this paper, the ideas of *Apriori* association rule mining in the field of data mining is used, and the level-wise scan algorithm and valid support count statistics are presented. Compared with the previous algorithms, this algorithm involves less computational time and is more accurate. It can be widely applied to many kinds of information processing fields, such as Automatic Indexing, Automatic Classification, Information Retrieval, Automatic Summarization.

## References

1. Wang, Y.-C.: Chinese Information Processing Technology. Press of Shanghai Jiao Tong University, Shanghai (1991)
2. Tan, H.-Y.: Research on Method of Automatic Recognition of Chinese Place Name based on Transformation. Journal of Software 12(11), 1608–1613 (2001)

3. Nie, J.-Y.: Unknown Word Detection and Segmentation of Chinese using Statistical and Heuristic Knowledge. *Communications of COLIPS* 5(1&2), 47–57
4. Ling, G.-C., Asahara, M., Matsumoto, Y.-J.: Chinese Unknown Word Identification Using Character-based Tagging and Chunking. In: Dignum, F.P.M. (ed.) *ACL 2003. LNCS (LNAI)*, vol. 2922, pp. 197–200. Springer, Heidelberg (2004)
5. Cui, S.-Q., Liu, Q., Meng, Y.: New Word Detection Based on Large-Scale Corpus. *Journal of Computer Research and Development* 43(5), 927–932 (2006)
6. Huang, X.-J., Wu, L.-D., Wang, W.-X., Ye, D.-J.: A Machine Learning Based Word Segmentation System without Manual Dictionary. *Pattern Recognition and Artificial Intelligence* 9(4), 297–303 (1996)
7. Luo, S.-F., Sun, M.-S.: Chinese Word Extraction Based on the Internal Associative Strength of Character Strings. *Journal of Chinese Information Processing* 17(3), 9–14 (2003)
8. Liu, T., Wu, Y., Wang, K.-Z.: A Chinese Word Automatic Segmentation System Based on String Frequency Statistics Combined with Word Matching. *Journal of Chinese Information Processing* 12(1), 17–25 (1998)
9. Ren, H., Zeng, J.-F.: A Chinese Word Extraction Algorithm Based on Information Entropy. *Journal of Chinese Information Processing* 20(5), 40–90 (2006)
10. Han, K.-S., Wang, Y.-C., Chen, G.-L.: Research on Fast High2frequency Strings Extracting and Statistics Algorithm with no Thesaurus. *Journal of Chinese Information Processing* 15(2), 23–30 (2001)
11. Jiang, S.-H., Dang, Y.-Z.: Segmentation Algorithm for Chinese Text Based on Length Descending and String Frequency Statistics. *Journal of the China Society for Scientific and Technical Information* 25(1), 74–79 (2006)
12. Jin, X.-Y., Sun, Z.-X., Zhang, F.-Y.: A Domain-independent Dictionary-free Lexical Acquisition Model for Chinese Document. *Journal of Chinese Information Processing* 15(6), 33–39 (2001)
13. MA, Y.-H., Wang, Y.-C., Su, G.-Y.: A Fast Approach of Extracting Repeated String from Chinese Text. *Acta Electronica Sinica* 12(12), 2177–2179 (2002)
14. Liu, H.: A New Approach for Doma in New Words Detection. *Journal of the China Society for Scientific and Technical Information* 20(5), 17–23 (2006)
15. Han, J.-W., Kamber, M.: *Data Mining Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco (2001)

# Using Lexical Patterns for Extracting Hyponyms from the Web\*

Rosa M. Ortega-Mendoza, Luis Villaseñor-Pineda, and Manuel Montes-y-Gómez

Laboratorio de Tecnologías del Lenguaje,  
Instituto Nacional de Astrofísica, Óptica y Electrónica, México  
{rmortega,villasen,mmontesg}@ccc.inaoep.mx

**Abstract.** This paper describes a method for extracting hyponyms from free text. In particular it explores two main matters. On the one hand, the possibility of reaching favorable results using only lexical extraction patterns. On the other hand, the usefulness of measuring the instance's confidences based on the pattern's confidences, and vice versa. Experimental results are encouraging because they show that the proposed method can be a practical high-precision approach for extracting hyponyms for a given set of concepts.

## 1 Introduction

Linguistic resources such as dictionaries, gazetteers and ontologies have a broad range of applications in computational linguistics and automatic text processing [4]. This kind of resources provides significant knowledge about languages, but they are expensive to build, maintain and extend.

At present, most linguistic resources are manually constructed. They contain high precision entries but have very limited coverage. As a result, their usefulness is still restricted to certain domains or specific applications. In order to overcome this problem, recently many researchers have been working on semiautomatic methods for their construction. In particular, there is a special interest in the extraction of synonyms, antonyms and hyponyms from free text documents [6, 7, 5, 8, 9, 10].

This paper focuses on the extraction of hyponyms (*is-a* relations) from free text. Specifically, it proposes a pattern-based method for automatically acquiring hyponyms from the Web. This method mainly differs from previous approaches [5, 8, 9, 10] in that it only considers lexical information. That is, whereas previous methods make use of lexico-syntactic patterns, the proposed approach exclusively employs patterns expressed at lexical level.

Working at lexical level makes the method easily adapted to different languages, but also brings out some additional challenges. For instance, it is necessary to take into consideration a large number of patterns in order to compensate their poor generalization degree. The proposed method confronts this requirement by applying, on the one hand, a text mining technique that allows acquiring many lexical patterns from the Web, and on the other hand, an iterative process for evaluating the confidence of

---

\* Work done under partial support of CONACYT (project grants 43990 and 61335).

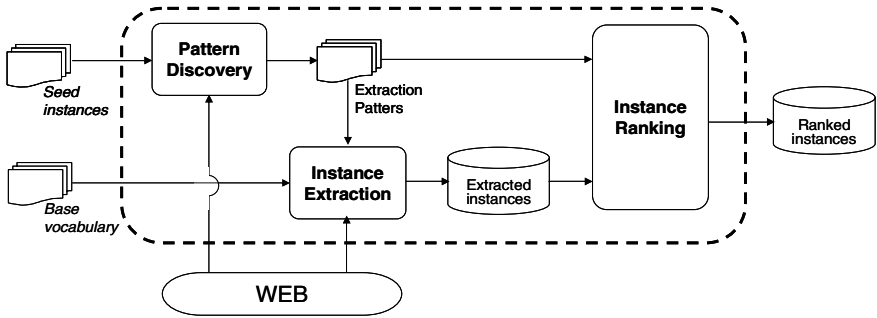


Fig. 1. General scheme of the proposed method

the discovered instances (pairs of hyponym-hypernym) to belong to the target relation. This process is supported on the assumption that pertinent instances are extracted by different patterns, and that valuable patterns allow extracting several pertinent instances. This process adopts some ideas proposed elsewhere [9], but modifies the computation of the initial weights of the acquired patterns. This modification allows assigning major importance to those patterns showing a good balance between precision and recall.

The following sections describe the proposed method and present some experimental results on the extraction of hyponyms related to a given base vocabulary.

## 2 Method at a Glance

Figure 1 shows the general scheme of the proposed method. It consists of three main modules: pattern discovery, instance extraction and instance ranking. The following paragraphs briefly describe the purpose and functionality of each module.

**Pattern discovery.** This module focuses on the discovery of a set of lexical extraction patterns from the Web. Its objective is to capture most of the writing conventions used to introduce a hyponym relation between two words.

This module adopts the method described in [2]. It mainly uses a small set of seed instances (pairs of hyponym-hypernym such as *apple-fruit*) to collect from the Web an extended set of usage examples of the hyponym relation. Then, it applies a text mining method [3] on the collected examples in order to obtain all maximal frequent word sequences<sup>1</sup>. These sequences express lexical patterns highly related to the hyponym relation. Finally, it retains only the patterns satisfying the following regular expressions:

$$\begin{aligned} &\langle \text{left-frontier-string} \rangle \text{ HYPONYM } \langle \text{center-string} \rangle \text{ HYPERNYM} \\ &\text{HYPERNYM } \langle \text{center-string} \rangle \text{ HYPONYM } \langle \text{right-frontier-string} \rangle \end{aligned}$$

As it can be seen, the Web is a very valuable resource for this step; however, it restricts the method to applications that do not require constructing the hyponym

<sup>1</sup> A maximal frequent word sequence is a sequence of words that occurs more than a predefined threshold and that is not a subsequence of another frequent sequence.

catalog on the fly. In our particular case, the discovery of the extraction patterns as well as the entire construction of the hyponym catalog are defined as off-line processes. Therefore, the quality of the resultant resource is much more relevant than the efficiency of its construction.

**Instance extraction.** In this module, the patterns discovered in the previous stage are applied over a target document collection in order to locate several text segments that presumably contain an instance of the hyponym relation. The result is a set of candidate hyponym-hypernym pairs.

To locate as many as possible hyponym relations our implementation of the module considers using the Web as target document collection. In addition, in order to extract as many as possible correct relations, it uses a user-given vocabulary to instantiate the patterns (i.e., to construct the Web queries).

For instance, having the pattern “*the HYPONYM is one of the HYPERONYM*” and the target concept *stone*, our method constructs the query “*the HYPONYM is one of the stones*”. Using the instantiated pattern it is possible to extract the hyponym-hypernym pair *diamond–stone* from the snippet “the diamond is one of the stones associated with Aries and Leo...”, but also it is possible to discover incorrect instances such as *privatization–stone* (from the snippet “the privatization is one of the stones of market development of regional economy...”). Therefore, to differentiate between right and wrong instances it is necessary to incorporate an extra module for evaluating and ranking patterns and instances.

**Instance ranking.** This module evaluates the confidence of the extracted instances to belong to the hyponym relation. Its purpose is to rank the instances in such a way that those with higher probability of being correct locate at the very first positions.

This evaluation bases on the idea that pertinent instances are extracted by different patterns, and that valuable patterns allow extracting several pertinent instances. In particular, we defined an *iterative evaluation process* where instance’s confidences are calculated based on pattern’s confidences, and vice versa.

The following section gives details on the iterative evaluation process; especially it defines the evaluation functions used to estimate the confidence of instances and patterns.

### 3 Iterative Evaluation Process

As we mentioned before, the iterative evaluation process calculates the confidence of instances and patterns in accordance with their association: i.e., an instance has a greater probability of being correct if it is associated to (was extracted by) several confidence patterns, and a pattern is more relevant if it is associated to (allow extracting) several confidence instances.

The most direct approach for measuring the confidence of a pattern (its association with the extracted instances) is through its precision and recall<sup>2</sup>. However, these measures are impossible to assess due to the lack of information on the extension of

---

<sup>2</sup> *Precision* indicates the percentage of correct instances extracted by the pattern. On the other hand, *recall* is the percentage of all relevant instances (in the target document collection) that were actually extracted by the pattern.

the relation at hand. In other words, it is impossible to know in advance the whole set of pairs hyponym-hypernym existing at the Web. Therefore, it is common to evaluate the association degree between patterns and instances (hyponym-hypernym pairs) using a pointwise mutual information metric [1]. In particular, we consider three different well-known metrics to compute the mutual information between a pattern  $p$  and an extracted instance  $i = (x, y)$ :

$$pmi_1(p, i) = \log \frac{P(x, p, y)}{P(*, p, *)P(x, *, y)} \quad (1)$$

$$pmi_2(p, i) = \frac{|x, p, y|}{|x, *, y|} \quad (2)$$

$$pmi_3(p, i) = \log \frac{|x, p, y| |*, p, *|}{|x, p, *| |*, p, y|} \quad (3)$$

where  $|x, p, y|$  and  $P(x, p, y)$  indicate the absolute and relative frequencies of the pattern  $p$  instantiated with terms  $x$  and  $y$ , and the asterisk (\*) represents a wildcard.

Based on any given association metric, we compute the confidence values of patterns and instances as proposed by [9]:

$$c_\pi(p) = \frac{\sum_{i \in I} \left( \frac{pmi(p, i)}{\max_{pmi}} \times c_\sigma(i) \right)}{|I|} \quad (4)$$

$$c_\sigma(i) = \frac{\sum_{p \in P} \left( \frac{pmi(p, i)}{\max_{pmi}} \times c_\pi(p) \right)}{|P|} \quad (5)$$

where  $c_\pi(p)$  and  $c_\sigma(i)$  are the confidences of pattern  $p$  and instance  $i$  respectively,  $\max_{pmi}$  indicates the maximum pointwise mutual information between all patterns and all instances,  $I$  is the set of instances extracted by pattern  $p$ , and  $P$  is the set of patterns that extract the instance  $i$ .

### 3.1 Computing the Initial Confidence of Patterns

It is noticeable from formulas (4) and (5) that the confidence values of patterns and instances are recursively defined. In this scheme, the initial confidence of patterns is commonly estimated by (4) using  $c_\sigma(i) = 1$  for the manually supplied seed instances.

Given that the set of seed instances is –in the majority of the cases– very small, this kind of “probabilistic” estimation tends to favor patterns with very high precision or very high recall, but not necessarily gives preferentiality to those patterns showing a good balance between both measures. In order to achieve this balance we propose to compute the initial confidence of patterns using a kind of  $F$ -measure metric<sup>3</sup>:

$$c_\pi(p) = \frac{F(p)}{\max_{\forall l \in P} \{F(l)\}}, \text{ where} \quad (6)$$

<sup>3</sup>  $F$ -measure is the weighted harmonic mean of precision and recall.

$$F(p) = \frac{2 \times E(p) \times R(p)}{E(p) + R(p)} \tag{7}$$

Here,  $E(p)$  indicates the proportion of seed instances extracted by pattern  $p$  (i.e., a kind of precision of  $p$ ),  $R(p)$  is the quotient between the number of seed instances and the whole set of instances extracted by  $p$  (i.e., a kind of recall of  $p$ ), and finally  $\max\{F(l)\}$  is a normalization factor.

## 4 Experimental Results

To evaluate the proposed method we approached the discovery of hyponyms for a given set of concepts in *Spanish* language. The following passages present the achieved results at each module of the method.

For pattern discovery, we considered a set of 25 seed instances and used Google to retrieve 500 text segments per seed instance. This way, we constructed a corpus of 12,500 segments expressing the hyponym relation. From this corpus we extracted 43 lexical extraction patterns. Some of these patterns are shown in table 1. It is noticeable that the quality of the discovered patterns is very diverse. Some are too specific and precise but not so applicable, whereas some others are too general, but guarantee a high coverage.

**Table 1.** Some lexical patterns for extracting hyponyms

<i>Original extraction patterns (in Spanish)</i>	<i>Extraction patterns (English translation)</i>
<i>el HYPONYM es un HYPERNYM que</i>	<i>the HYPONYM is a HYPERNYM that</i>
<i>el HYPONYM es el único HYPERNYM</i>	<i>the HYPONYM is the single HYPERNYM</i>
<i>el HYPONYM es uno de los HYPERNYM mas</i>	<i>the HYPONYM is one of the HYPERNYM more</i>
<i>las HYPONYM son una HYPERNYM</i>	<i>the HYPONYM are a HYPERNYM</i>
<i>El uso de la HYPONYM como HYPERNYM</i>	<i>the use of HYPONYM as HYPERNYM</i>

For instance extraction, we firstly instantiated the acquired patterns using a given set of concepts (hypernyms). In particular, we considered the following five terms: *banco* (bank), *enfermedad* (disease), *felino* (feline), *profesión* (profession) and *roca* (stone). The selection of these terms allow studying the use of the patterns in very different topics, and therefore to obtain a general notion about their applicability. Using the instantiated patterns as web queries, we extracted 851 candidate hyponym instances: 193 related to bank, 307 to disease, 9 to feline, 226 to profession, and 116 to stones.

Finally, we ranked the list of hyponyms by applying the evaluation process described in section 3. In a first experiment, we compared the results obtained by using three different well-known metrics for computing the mutual information between a pattern and an instance (see formulas 1-3). Figure 2 shows the achieved precision curves after three iterations. It is clear that  $pmi_1$  is the best approach for this specific problem. In average, it was 50% points better than  $pmi_2$  and 30% superior to  $pmi_3$ . It is important to mention that in this experiment the initial confidence of patterns were estimated as usually, that is, using  $c_{\mathcal{A}}(i) = 1$  for the manually supplied seed instances.



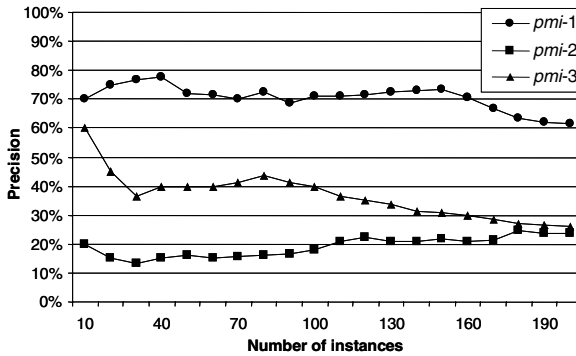


Fig. 2. Comparing three pointwise mutual information metrics

In a second experiment, we evaluated the impact of applying the proposed method for computing the initial confidence of patterns (refer to formula 4). The following figures present the evaluation results on the 200-top ranked instances. On the one hand, figure 3 shows the results obtained by the proposed method for the first three iterations of the evaluation process. In this case, we used the *F*-measure metric to compute the initial pattern's confidences. On the other hand, figure 4 compares the

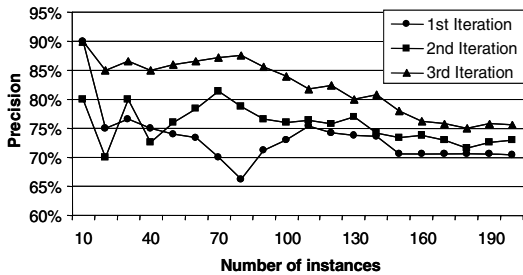


Fig. 3. Results of the proposed method

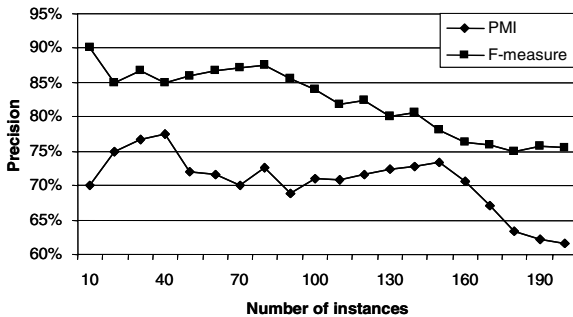


Fig. 4.  $Pmi_1$  vs. *F*-measure for computing the initial pattern confidences

precision curves after three iterations using two different metrics for computing the initial confidences, the traditional one based on the  $pmi_i$  metric and the proposed one based on the  $F$ -measure. The achieved results corroborate the relevance of the idea of iteratively computing the confidence of instances and patterns in accordance with their association. In addition, they show the impact of the initial pattern's confidences over the final ranking, and demonstrate the convenience of giving more importance to those patterns showing a good balance between precision and recall.

## 5 Conclusions

This paper proposed a new method for extracting hyponyms from free text. The method consists of three main modules. The first one focuses on the acquisition of extraction patterns from the Web, the second one uses the acquired patterns to extract a set of instances that presumably belong to the hyponym relation, finally, the third module considers the ranking of the extracted instances.

In particular, the proposed method differs from previous approaches in that: (i) it only considers lexical information, whereas the rest of the works make use of lexico-syntactic patterns, (ii) it uses all discovered patterns –specific and general– for instance extraction, and (iii) it applies a new metric, based on  $F$ -measure, for computing the initial confidence of the extraction patterns.

The presented experimental results demonstrated the feasibility of using lexical patterns to extract hyponyms from free texts as well as the pertinence of the proposed metric for computing the initial pattern's confidence.

Future work will be focused on concluding about the language independence and large-scale performance of the proposed method. In particular, our current results have demonstrated that the method is adequate for Spanish (a language with moderate complex morphology), therefore, we expect that it will be also useful for dealing with other romance languages or other languages with relative simple morphology such as English. However, it is very possible that the method will not be pertinent for dealing with languages having a complex morphology, for instance, agglutinative languages such as German or Arabic.

On the other hand, we plan to use the method to extract hyponyms for a large number of concepts (i.e., using a greater base vocabulary). Using a large number of concepts will not be a problem for the method; on the contrary, we believe that having more information will allow to obtain an accurate estimation of the confidences of instances and patterns.

## References

1. Blohm, S., Cimiano, P.: Learning Patterns from the Web - Evaluating the Evaluation Functions - Extended Abstract. In: OTT 2006. Ontologies in Text Technology: Approaches to Extract Semantic KnowledgeInformation, Osnabrück, Germany (2006)
2. Denicia, C., Montes, M., Villaseñor, L., García, R.: A Text Mining Approach for Definition Question Answering. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) FinTAL 2006. LNCS (LNAI), vol. 4139, Springer, Heidelberg (2006)

3. García-Hernández, R., Martínez-Trinidad, F., Carrasco-Ochoa, A.: A New Algorithm for Fast Discovery of Maximal Sequential Patterns in a Document Collection. In: International Conference on Computational Linguistics and text Processing, CICLing-2006. Mexico City, Mexico (2006)
4. Gelbukh, A., Sidorov, G.: Procesamiento automático del español con enfoque en recursos léxicos grandes. In: IPN, p. 240 (2006)
5. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: Conference on Computational Linguistics (COLING-1992), Nantes, France (1992)
6. Lin, D., Zhao, S., Qin, L., Zhou, M.: Identifying synonyms among distributionally similar words. In: International Joint Conference of Artificial Intelligence (IJCAI-2003). Acapulco, Mexico (2003)
7. Lucero, C., Pinto, D., Jiménez, H.: A Tool for Automatic Detection of Antonymy Relations. In: Workshop on Herramientas y Recursos Lingüísticos para el Español y el Portugués. IBERAMIA-2004. Puebla, Mexico (2004)
8. Mann, G.S.: Fine-Grained Proper Noun Ontologies for Question Answering. SemaNet-02: Building and Using Semantic Networks. Taipei, Taiwan (2002)
9. Pantel, P., Pennacchiotti, M.: Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In: Conference on Computational Linguistics/Association for Computational Linguistics (COLING/ACL-2006), Sydney, Australia (2006)
10. Ravichandran, D., Pantel, P., Hovy, E.: The Terascale Challenge. In: Proceedings of KDD Workshop on Mining for and from the Semantic Web. Seattle, WA, USA (2004)

# On the Usage of Morphological Tags for Grammar Induction\*

Omar Juárez Gambino and Hiram Calvo

Center for Computing Research, National Polytechnic Institute,  
Av. Juan de Dios Bátiz s/n, esq. Av. Othón de Mendizábal, México, D.F., 07738. México  
omarjgb06@sagitario.cic.ipn.mx, hcalvo@cic.ipn.mx  
www.hiramcalvo.com

**Abstract.** We present a study on the effect of adding morphological tags to the training corpus of a grammar inductor. For this purpose, we carried out several experiments using the grammar induction system called Alignment-Based Learning (ABL) and the CAST-3LB syntactically tagged Spanish corpus for training and testing. ABL produces a set of possible constituents with a word alignment process. We developed an algorithm which converts the hypotheses generated by ABL into ordered production rules. Then our algorithm groups them into possible phrase groups (constituents). These phrase groups correspond to the syntactic tagging of the unannotated text. We compared the phrase groups obtained by our algorithm with the manually tagged groups of CAST-3LB. The experiments in the grammar induction process consisted on trying three different variants for the training corpus: (1) using words; (2) using only the morphological tags; and (3) adding morphological tags to words. Our experiments show that the inclusion of morphological tags in the grammar induction process improves significantly the performance of ABL.

**Keywords:** Grammar Induction, Syntactic Tagging, Morphological Tags, ABL.

## 1 Introduction

In general a syntactic parser needs handwritten rules; this is a costly task and it is very difficult to cover all the language phenomena. Trying to write all the rules for a syntactic parser is equivalent to describe the I-language as defined by Chomsky [6] — I-language is the mentally represented linguistic knowledge. There are many techniques to find these rules automatically, but generally they use syntactically tagged corpora for learning. Consider, for example, the Charniak syntactic-probabilistic parser [2]. This parser is trained using the Penn Treebank manually tagged corpus and achieves 90.1% average precision/recall for sentences of length  $\leq 40$  and 89.5% for sentences of length  $\leq 100$ . In contrast, our intention is to focus on improving the process of finding structural regularities in untagged texts which might allow to obtain the main constituents of sentences just like a syntactic parser.

---

\* Work done under support of the Mexican Government (CONACYT, SNI, PIFI, and SIP-IPN).

Grammar inference consists on learning the grammar structure from data [3]. These methods can be classified in two different approaches, supervised and unsupervised. In general the performance of supervised methods is greater than unsupervised methods, but they need a structured training corpus and their domain is limited. On the other hand, unsupervised methods do not require a structured training corpus and their domain is not limited (*i.e.*, it can be any domain depending on the training corpus). In this paper we use an unsupervised grammar inductor. For this purpose, we trained the ABL grammar inductor [10]. ABL has reported to work for inducing grammars for the ATIS [7], OVIS [1] and WSJ [7] corpora. Our goal is to evaluate the impact of adding morphological tags in the training corpus. In the following sections we will explain our method in detail.

## 2 ABL

The grammar inductor used in our experiments was ABL. The algorithm used by ABL is based in Harris's notion of substitutability [5] which states that *constituents of the same type can be replaced by each other*. ABL reverses Harris's idea [10]: *if (a group of) words can be replaced by another, then both (group of) words are constituents of same type*. ABL consists of three phases:

- **Alignment phase.** This phase compares all the sentences in the input corpus to each other. When the algorithm finds a match on identical words in a pair of sentences, it takes the distinct parts of the sentences as possible constituents (hypothesis), then it marks this hypothesis with a number to identify it (Figure 1a).
- **Clustering phase.** In this phase the hypotheses that share the same context are grouped (Figure 1b).
- **Selection learning phase.** This phase chooses the best hypothesis when there exists overlapping between two hypotheses. The selection is based in probabilistic methods; see overlapping of hypotheses 1 and 2 in Figure 1c.

a) the boy [plays with the ball]<sub>1</sub>    b) the boy [plays with the ball]<sub>1</sub>  
       the boy [runs in the park]<sub>2</sub>        the boy [runs in the park]<sub>1</sub>

c) [He plays]<sub>1</sub> in the park  
       [~~She~~ (~~runs~~)<sub>1</sub> in the park]<sub>2</sub>  
       She (ate a hamburger)<sub>2</sub>

Fig. 1. ABL phases

Some parameters can be configured for the previously mentioned ABL phases [4], we explain them below.

### Alignment phase methods

- **Default.** Uses a cost function to find the longest common subsequences in two sentences.
- **Biased.** This method biases the cost function towards linking words that have a similar relative offset in the sentence.
- **All.** Introduces all possible alignments, no-probabilistic cost function is used.

### Selection learning phase methods

- **First.** This is a non-probabilistic method; it assumes that a hypothesis that has been learnt before is correct and it does not change it.
- **Leaf.** This method computes the probability of a hypothesis by counting the number of times that the particular words of this hypothesis have occurred in the learned text as other hypothesis. This probability is then divided by the total number of hypotheses.
- **Branch.** This method computes the same probability as the Leaf method but in addition it considers the counts of hypotheses' labels to keep the most assigned label.

We selected the Biased alignment method and the Branch selection learning method for our experiments. See details in Section 4.

## 3 Methodology

We developed an algorithm to adapt the ABL output to a bracketing scheme that would allow us to compare with a manually tagged corpus. In this section we describe the process we developed to obtain phrase groups from the output that ABL provides (see Figure 2).

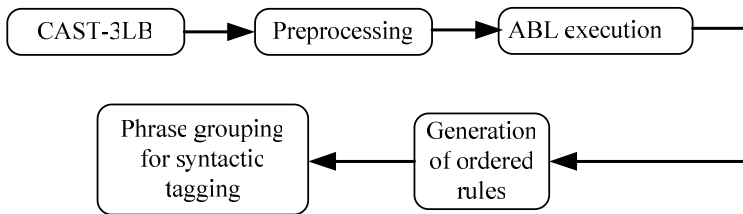


Fig. 2. Syntactic Tagging process

### 3.1 Preprocessing

The first step for our experiments consists on generating three variants of the CAST-3LB<sup>1</sup> [8] corpus. In the first variant we used only words (Figure 3a); the second

<sup>1</sup> CAST-3LB is part of the 3LB project, financed by the Science and Technology Ministry of Spain. 3LB, (FIT-150500-2002-244 and FIT 150500-2003-411).

- a) desde entonces entró en silencio absoluto .  
 b) sps00 rg vmis3s0 sps00 ncms000 aq0ms0 Fp  
 c) desde/sps00 entonces/rg entró/vmis3s0 en/sps00 silencio/  
 ncms000 absoluto/aq0ms0 ./Fp

**Fig. 3.** Three Variants of CAST-3LB corpus for the sentence ‘since then (he/she) remained in complete silence’

variant consists on only morphological tags (Figure 3b); and the third one consists on both morphological tags and words (Figure 3c).

### 3.2 ABL Execution

The ABL algorithm produces a labeled, bracketed version of the corpus, with an identification number for each constituent and its location in the sentence (see Figure 4).

las reservas en oro se valoran en\_base\_a 300\_dólares  
 estadounidenses por cada onza troy de oro .

@@@(0,1,[3])(15,16,[2])(2,3,[152])(8,9,[154])(0,16,[0])(13,14,[280])(14,15,[281])(9,10,[285])(4,5,[288])(3,4,[292])(1,2,[297])(10,11,[302]) **(11,13,[305])** (3,8,[153]) (10,13,[291]) (9,15,[155])

**Fig 4.** ABL output for the sentence ‘gold reserves are valued by 300 U.S. dollar for each troy ounce’

Each set of numbers in parenthesis represents a constituent. The first number in parenthesis is the starting point of the constituent in the sentence; the second one is the final position; and the number in brackets identifies the hypothesis. For example the hypothesis 305 (see Figure 4) consists on the words beginning in position 11 to 13: *onza troy*.

We use the output produced by ABL to obtain the words which were grouped as constituents, but with this representation the grouping order of the sentence is not evident. The grouping order of a sentence is needed to reconstruct the sentence and find the relationships between constituents.

### 3.3 Generation of Ordered Rules

We developed Algorithm 1 (Figure 5) to obtain ordered production rules. We define a non-terminal as a constituent which is composed by other constituents. The first step in this algorithm consists on sorting the constituents by production order. This algorithm uses the final position, the length, and the initial position of the sentences for this process. The algorithm stores the components of the non-terminal constituents. See Figure 7b for an example of the output of Algorithm 1.

```

func Get_ProductionRules(cons_lis) #The list of constituents
  var cons_ord: matrix of n x 2 constituents
      constituents: hash table of non-terminal constituents
      tot_cons, i, j, k, ban: Integers
  sorted_cons = Sort_Cons(cons_lis) #Sort constituents list
  tot_cons = number of items of sorted_cons
  for (i=tot_cons; i>=0; i--)
    ban=0
    j=i
    k=j+1
    while (ban==0)
      if ((sorted_cons[j][0]>=sorted_cons[k][0])
          and (sorted_cons[j][1]<=sorted_cons[k][1]))
        constituents{sorted_cons[k][0]} += sorted_cons[j][0]
      else
        k= k +1
      end if
    end while
  end for
end func

```

**Fig. 5.** Algorithm 1. It converts ABL hypotheses to ordered production rules

### 3.4 Syntactic Tagging Algorithm

The next step is Algorithm 2 shown in Figure 6. Algorithm 2 receives a non-terminal constituent as parameter and produces a new constituent expanding the ordered production rules by a recursive method. When the recursive method begins, it evaluates if the constituent is composed by other non-terminal constituents, that being the case, the algorithm is called again with these non-terminal constituents; otherwise it returns the words contained in the constituents.

This algorithm stops when all the constituents of the sentence have been evaluated. It is important to identify when a constituent is a non-terminal, because this means that all the constituents composing the non-terminal are grouped in a phrase. We put these constituents in parenthesis to manifest that they form a phrase (see Figure 7c).

```

func Generate_Phrase(non-terminal_cons)#Non-terminal constituent
  #If the constituent is a non-terminal
  if (exists constituents{non-terminal_cons})
    print "(" #It is a phrase
    for each cons (constituents{non-terminal_cons})
      Generate_Phrase(cons)
    end for
    print ")" #Ends the phrase
  else #It is a terminal constituent
    print constituents{non-terminal_cons}
  end if
end func

```

**Fig. 6.** Algorithm 2. It expands the production rules



In Figure 7b we can see that the constituent *mayo\_de\_1998* ‘may of 1998’ was not identified by ABL. It means that ABL did not produce any constituent that contains this word. The reason of this situation is because in the selection learning phase that word was eliminated to avoid a hypothesis overlapping. Algorithm 1 solves this problem by inserting orphan words between their siblings.

**a) ABL output**  
 desde mayo\_de\_1998 era vocal asesora de la  
 dirección\_general\_de la tesorería\_general\_de la seguridad\_social.  
 @@@(0,9,[0])(5,6,[564])(6,7,[1126])(8,9,[3240])(0,1,[2947])  
 (7,8,[2048])(2,3,[21454])(3,5,[21457])(0,2,[21453])(5,7,[1713])

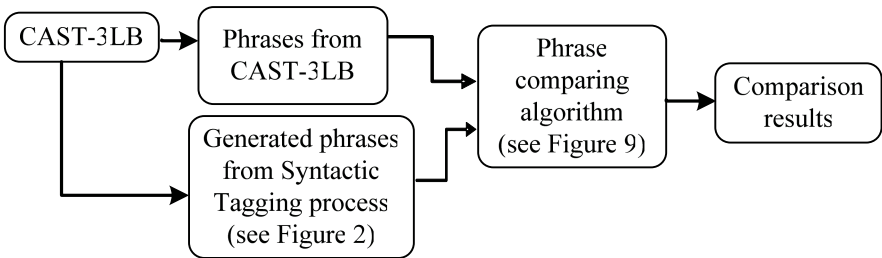
**b) Ordered Rules process (Output of Algorithm 1)**  
 [0] → [21453] [21454] [21457] [1713] [2048] [3240]  
 [1713] → [564] [1126]  
 [21453] → [2947] [mayo\_de\_1998]

**c) Grouping Phrases process (Output of Algorithm 2)**  
 ((desde mayo\_de\_1998) era (vocal asesora) (de la)  
 dirección\_general\_de la tesorería\_general\_de la seguridad\_social.))

**Fig. 7.** Example of result of the Syntactic Tagging algorithm for the sentence ‘since may of 1998, (she) was the advisor member of the head office of the general office of the treasurer of social security’

### 4 Experiments and Results

We followed the process described in Figure 8 to carry out the experiments. We used the CAST-3LB syntactically tagged Spanish corpus which is composed approximately by 100,000 words and 3500 sentences. We can see an example of a



**Fig. 8.** Phrase comparing process

```

(S
  (sadv-MOD
    (rg Tampoco tampoco))
  (gv
    (vmip1p0 recordamos recordar))
  (S.F.C-CD
    (sp-CC
      (prep
        (sps00 por por))
      (sn
        (grup.nom.s
          (pt0cs000 qué qué))))
  (sn.e-SUJ *0*)
  (gv
    (vmis3p0 llegaron llegar)))
(Fp . .) ...

```

**Fig. 9.** Tagged sentence from the CAST-3LB corpus ‘we neither remembered the reason why they arrived’

CAST-3LB corpus sentence in the in Figure 9, the parentheses shown in this example are used to evaluate Algorithm 1 (more details in Section 4.2). We executed ABL using the following parameters:

- **Alignment phase.** Alignment method: Biased.
- **Selection learning phase.** Selection method: Branch.

We tested other parameters and we found the best results by selecting these parameters. In addition we used the ABL clustering process in all the experiments.

#### 4.1 Metrics

We used three metrics to evaluate the results of the experiments. The metrics indicate how similar group phrases are.

- **Recall.** This metric shows how many correct phrases were generated. It is the percentage of the correct phrases that are found in the CAST-3LB corpus.

$$\text{Recall} = \frac{\sum_{s \in \text{sentences}} | \text{Phrase\_Compare}(\text{generated\_3LB}(s), \text{generated\_ABL}(s)) |}{\sum_{s \in \text{sentences}} | \text{generated\_3LB}(s) |} \quad (1)$$

- **Precision.** This metric indicates how many generated phrases were correct. It is the percentage of the correctly generated phrases with respect to all generated phrases.

$$\text{Precision} = \frac{\sum_{s \in \text{sentences}} | \text{Phrase\_Compare}(\text{generated\_3LB}(s), \text{generated\_ABL}(s)) |}{\sum_{s \in \text{sentences}} | \text{generated\_ABL}(s) |} \quad (2)$$

- **F-score.** It combines the recall and precision measure into one score. In our experiments Precision and Recall are equally important, so  $\beta$  is set to 1.

$$F_{\beta} = \frac{(\beta^2 + 1) * \text{Precision} * \text{Recall}}{(\beta^2 * \text{Precision} ) + \text{Recall}} \quad (3)$$

## 4.2 The Phrase Comparing Algorithm

Algorithm 3 (Figure 10) compares the phrases generated with Algorithm 2 and the manually tagged phrases of the CAST-3LB corpus (see Figure 9). All the phrases of the CAST-3LB corpus and the phrases obtained by Algorithm 2 are in parentheses. Algorithm 3 measures the coincidence of opening and closing between these parentheses. When a coincidence is found, this means that a phrase has matched. For example, in Figure 11 we show an example of the comparison between two phrases; we can see some coincidences in both first and third opening parentheses, and in both

```
func Phrase_Compare(a, b) #a and b are sentences
var
  pos_a = 0, pos_b = 0, match = 0, fail= 0
  car_a, car_b: pair of characters
while (pos_a < length(a) or pos_b < length(b))
  if (car_a ∈ a == "(")
    move in the sentence and increment pos_a while car_a = "("
    if (car_b ∈ b == "(")
      move in the sentence and increment pos_b while car_b = "("
      match = match + 1 #Phrase match found
    else
      fail = fail + 1 #Phrase match failed
    end if
  else
    if (car_a ∈ a == ")")
      move in the sentence and increment pos_a while car_a = ")"
      if (car_b ∈ b == ")")
        move in the sentence and increment pos_b while car_b = ")"
        match = match + 1 #Phrase match found
      else
        fail = fail + 1 #Phrase match failed
      end if
    end if
  end while
end func
```

**Fig. 10.** Algorithm 3. It compares manually tagged phrases from CAST-3LB vs. phrases from Syntactic Tagging Algorithm

CAST-3LB	(tampoco recordamos((por qué)llegaron).)
Our Algorithm	(tampoco (recordamos(por qué)llegaron.))

**Fig. 11.** Example of the Phrase Comparing process for the sentence ‘we neither remembered the reason why they arrived’

first and third closing parentheses, but we can also see an extra parenthesis in the second output that does not match. With the results of these measures we applied the metrics previously defined.

### 4.3 Results

We carried out experiments with 3 the variants of the CAST-3LB corpus. In experiment 1 we used only words; in experiment 2 we used only morphological tags; and in experiment 3 we added morphological tags to word. Table 1 shows the results of the experiments.

**Table 1.** Results of our experiments

<b>Experiment</b>	<b>Recall</b>	<b>Precision</b>	<b>F-score</b>
(1) Words only	19.35%	32.19%	24.17%
(2) Morphological tags	19.83%	30.97%	24.18%
(3) Words + Morphological tags	<b>28.13%</b>	<b>38.34%</b>	<b>32.45%</b>

Adding Morphological tags to words improves the performance in all metrics. Compared with supervised methods (such as the Charniak parser [2]), these results may seem low; however, it should be considered that we are using a completely unsupervised method.

**Table 2.** Results of van Zaanen's experiments with the WSJ corpus

	<b>Recall</b>	<b>Precision</b>	<b>F-score</b>
Random baseline	23.94%	22.62%	23.27%
Upper limit	52.86%	100.00%	69.16%
Default ABL	12.56%	42.56%	19.26%

When compared with the results presented by van Zaanen [11] shown in Table 2, we can see that the results of our first two experiments are similar to those obtained by van Zaanen for 1094 sentences of the Wall Street Journal (WSJ) corpus. It is important to note that a direct comparison is not totally fair because both corpora are different, however they share common features such as the number of sentences (both are composed by more than 1000 sentences), and the sentences length (in average, 30 words per sentence in both corpora).

## 5 Conclusions and Future Work

We have presented a study on the effect of adding morphological tags to the training corpus of a grammar inductor. We found that combining words with morphological tags yields a better performance of the grammar induction process; we believe that this is mainly because adding morphological tags helps by disambiguating words, thus, narrowing the search space for alignments which in turn improves the alignment and selection learning phases of ABL.

Furthermore, the addition of morphological tags suggest that better results can be achieved for medium size corpora, improving the original results shown by van Zaanen [11]; we shall experiment in the future with corpora in other languages such as the Wall Street Journal.

Future work may involve adding information for disambiguation at deeper levels, such as information on word senses to evaluate its effects in grammar induction.

## References

1. Bonnema, R., Bod, R., Scha, R.: A DOP model for semantic interpretation. In: Proceedings of the Association for Computational Linguistics/European Chapter of the Association for Computational Linguistics, Madrid, pp. 159–167 (1997)
2. Charniak, E.: A Maximun-Entropy-Inspired Parser. In: Proceedings of NAACL-2000 (2000)
3. Dupont, P.: Grammatical Inference: Formal and Heuristics Methods. Carnegie Mellon University (1997)
4. Geertzen, J., van Zaanen, M.: Alignment-Based Learning Reference Guide. Thecnical Report, Macquarie University (2006)
5. Harris, S.Z.: Structural Linguistic. University of Chicago Press, Chicago (2000)
6. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. MIT Press, Cambridge (2000)
7. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a Large Annotaded Corpus of English: The Penn Treebank. *Computational Linguistics* 19(2), 313–330 (1993)
8. Navarro, B., Civit, M., Antonia Martí, M., Marcos, R., Fernández, B.: Syntactic, semantic and pragmatic annotation in Cast3LB. In: *Shallow Processing of Large Corpora (SProLaC)*, a Workshop of Corpus Linguistics, Lancaster, UK (2003)
9. van Zaanen, M., Adriaans, P.: Alignment-Based Learning versus EMILE: A Comparison. In: Krose, B., de Rijke, M., Schreiber, G., van Someren, M. (eds.) *Proceedings of the Belgian-Dutch Conference on Artificial Intelligence (BNAIC)*, Amsterdam, The Netherlands, pp. 315–322 (October 2001)
10. van Zaanen, M.: ABL: Alignment-Based Learning. In: *COLING 2000*, pp. 961–967 (2000)
11. van Zaanen, M.: *Bootstrapping Structure into Language: Alignment-Based Learning*. PhD Thesis, School of Computing, University of Leeds, U.K (2001)

# Web-Based Model for Disambiguation of Prepositional Phrase Usage\*

Sofía N. Galicia-Haro<sup>1</sup> and Alexander Gelbukh<sup>2</sup>

<sup>1</sup> Faculty of Sciences UNAM University City, Mexico City, Mexico  
sngh@ciencias.unam.mx

<sup>2</sup> Center for Computing Research, National Polytechnic Institute, Mexico  
gelbukh@cic.ipn.mx,  
www.Gelbukh.com

**Abstract.** We explore some Web-based methods to differentiate strings of words corresponding to Spanish prepositional phrases that can perform either as a regular prepositional phrase or as idiomatic prepositional phrase. The type of these Spanish prepositional phrases is preposition–nominal phrase–preposition (P–NP–P), for example: *por medio de* ‘by means of’, *a fin de* ‘in order to’, *con respecto a* ‘with respect to’. We propose an unsupervised method that verifies linguistic properties of idiomatic prepositional phrases. Results are presented with the method applied to newspaper sentences.

## 1 Introduction

There exist certain word combinations of the type preposition–nominal group–preposition (P–NP–P) that can be idiosyncratic in nature syntactically, or semantically, or both; we call them  $EXP_{PNP}$ . Automatic determination of such  $EXP_{PNP}$  groups can help in different tasks of natural language processing.

Spanish has a great number of prepositional phrases of the type P–NP–P more or less fixed. Among them: *a fin de* (in order to), *al lado de* (next to), *en la casa de* (in the house of), etc. The  $EXP_{PNP}$  (*a fin de*, *al lado de*) define three or more simple forms (since the nominal group can contain more of a simple form) as one lexical unit. Specifically, such combinations are frequently equivalent to prepositions, i.e., they can be considered as one multiword preposition: e.g., in order to is equivalent to ‘for’ (or ‘to’) and has no relation with order.

In opposition, regular P–NP–P ( $REG_{PNP}$ ) are analyzed considering the initial combination P–NP like a unit, and the second preposition as a one introducing a complement, not always linked to the preceding NP. Therefore, it is necessary to distinguish which of the Spanish P–NP–P should be analyzed as an  $EXP_{PNP}$  and which should be analyzed as a  $REG_{PNP}$ .

The PNP-EXP groups are used frequently in everyday language, therefore natural language applications need to be able to identify and treat them properly. Apart from syntactic analysis the range of applications where it is necessary to consider their

---

\* Work partially supported by Mexican Government (CONACyT, SNI, COFAA, SIP-IPN).

specific non compositional semantic is wide: machine translation, question answering, summarization, generation, etc.

There is no a complete compilation of the EXP<sub>PNP</sub> groups. [10] compiled the widest list but he himself considers that a prepositive relation study is something incomplete “susceptible of continuous increase”. In addition, the main Spanish dictionaries [6], [11] do not contain the information necessary for a computational resource. Even if we could compile an exhaustive EXP<sub>PNP</sub> list, they present different uses and their meaning agree with context.

In this work, we mainly investigated web-based methods to disambiguate the use of prepositional phrases as fixed forms from the literal ones, for example:

1. Idiomatic expression: *a fin de obtener un ascenso* ‘to obtain a promotion’ (literally ‘at end of’),
2. Part of a larger idiom: *a fin de cuentas* ‘finally’ (literally ‘at end of accounts’),
3. Free combination: *a fin de año obtendrá un ascenso* ‘at the end of the year she will be promoted’.

Identifying non-compositional or idiomatic multi-word expressions is an important task for any computational system, in recent years attention has been paid to practical methods for solving this problem [5], [1], and specially for prepositional phrases [13]. In this work, we analyze those linguistic properties of EXP<sub>PNP</sub> that differentiate them from REG<sub>PNP</sub> and idioms. For each property, we propose a web-based method that finally could contribute with a specific measure value in order to decide the nature of the P–NP–P phrase according to such linguistic properties.

In Section 2 we present the linguistic characteristics of the EXP<sub>PNP</sub>. In Section 3 we present the method by which we detect P–NP–P phrases and their context. Section 4 describes the association measures that we considered to disambiguate the three uses of P–NP–P. In Section 5, we present the obtained results.

## 2 Some Linguistic Characteristics of the EXP<sub>PNP</sub>

In this section we discuss some linguistics properties that place EXP<sub>PNP</sub> in a different group from the regular prepositional phrases and from idioms. In Spanish grammar, EXP<sub>PNP</sub> groups are denominated adverbial locutions [12] or prepositional locutions [10] according to their function. [12] indicates that locutions could be recognized by its rigid form that does not accept modifications and the noun that shows a special meaning; or by its global meaning, that is not the sum of the meanings of its components. For word combinations lexically determined that constitute particular syntactic structures, [9] indicate their properties: restricted modification, non composition of individual senses and the fact that nouns are not substitutable.

In [8] we present the linguistic properties of idiomatic P–NP–P Spanish phrases, we take in account the more general characteristics of them that are the following:

1. **Restricted modification.** Many of the nouns found in EXP<sub>PNP</sub> groups cannot be modified by an adjective. For example: *por temor a* (to avoid, literally: by fear of) vs. *por gran temor a* (by great fear of). In some cases, the modification forces

to take a literal sense of the prepositional phrase, for example in the following sentences:

- ... *por el gran temor a su estruendosa magia* (by the great fear to its uproarious magic)
  - ... *denegó hoy la libertad bajo fianza por temor a una posible fuga*. (today denied the freedom on bail to avoid a possible flight)
2. **Non substitutable nouns.** The noun inside the EXP<sub>PNP</sub> cannot be replaced by a synonym. For example in the phrase: *se tomará la decisión de si está a tiempo de comenzar la rehabilitación* (the decision will be taken on if it is the right time to begin the rehabilitation), where *a tiempo de* cannot be replaced by *a período de* (on period of), *a época de* (time of).
  3. **Part of idioms.** Some EXP<sub>PNP</sub> groups initiate fixed phrases or can be literal phrases according to the context, in addition to their use as idiomatic expression.

Examples:

“*al pie de*” (lit. ‘to the foot of’) It appears as idiomatic expression in:

*La multitud que esperó paciente al pie de la ladera de la sede de la administración del canal, corrió hacia arriba ...* (The patient multitude that waited at the base of the slope of the seat of the channel administration, run upwards ...)

“*al pie de*” It initiates a larger idiom: *al pie de la letra* (exactly) in:

*Nuestro sistema de procuración de justicia se ha transformado y en vez de observar al pie de la letra las leyes ...* (Our system of justice care has been transformed and instead of observing the laws exactly ...)

“*al pie de*” It initiates a free combination in:

*El anillo estaba junto al pie de María* (The ring was next to the foot of Maria)

### 3 Structure of the Prepositional Phrases and Their Context

We wrote a Perl program to automatically determine each P–NP–P and its right and left contexts. We used the AGME<sup>1</sup> tool to define the POS of each word and a very wide list of prepositions (P) obtained from [10] that includes prepositions with liberality.

The grammar defining the structure of the P–NP–P consists of the following rules:

PP → P NP P

NP → N | D N | V-Inf | D V-Inf

where PP stands for prepositional phrase, N for noun, D for determinant, and V-inf for infinitive verb (in Spanish, infinitives can be modified by a determinant: *el fumar está prohibido*, literally ‘the to-smoke is prohibited’).

<sup>1</sup> <http://www.cic.ipn.mx/~sidorov/agme/>



For example, in the following sentence the program detects six P–NP–P phrases (*de residencia a, a los inmigrantes para, a la falta de, de mano de, de obra en, según cálculos del*):

*El presidente cree necesario que el Gobierno agilice los permisos de residencia a los inmigrantes para poder responder a la falta de mano de obra en el país que, según cálculos del Departamento de Trabajo, impide cubrir unos 23,000 empleos en diversos sectores.*

To determine the right and left context of each P–NP–P for the analysis, we considered the following characteristics to delimit them, in the following order: (1) Punctuation, (2) Conjunctions, (3) Articles, adjectives and noun phrases.

For the same paragraph, the P–NP–P phrases and their reduced context were: *permisos de residencia a los inmigrantes, residencia a los inmigrantes para poder esponder, responder a la falta de mano, falta de mano de obra, mano de obra en el país, según cálculos del Departamento*. For the phrase (*mano de obra en el país*) the first cut was due to punctuation (,) the second cut was due to a conjunction (*que*) for the right context. For the left context, the article was the cut to leave a noun.

## 4 Web-Based Testing of Linguistic Properties

In this section we analyze the linguistic properties that could help to differentiate  $EXP_{PNP}$  groups from the  $REG_{PNP}$  and from idioms. For each property we try to find examples in Internet to decide if such property is accomplished.

We used the Internet as corpora. Internet was accessed using the Google search engine limited to Spanish language. Google statistics of co-occurrences of any two strings with any  $N$  intermediate words can be gathered by queries in quotation marks containing these strings separated with an asterisk, e.g., “*pie de \* página*”. The search engine returns phrases as: “*pie de cada página*”, “*pié de cada página*”, “*pie de lucha · Página*”, “*Pie de Página de la página*”, etc.

### 4.1 Restricted Modification

Since the nouns in  $EXP_{PNP}$  groups cannot be modified by an adjective we searched for possible noun phrase modifications in each  $P_1$ –NP– $P_2$  with its specific context, i.e., we searched for phrases of the type “context  $P_1$  \* NP \*  $P_2$  context”.

For example, if we have the  $EXP_{PNP}$  phrase “*con motivo de*” (that can be replaced by another single preposition: *por* ‘by’) and we want to find examples of restricted modification of a complete sentence containing it, we need to consider its specific context:

*Está bien que con motivo de fin de año todos tomen vacaciones, pero ¿todos a la vez?* (It is well that in occasion of end of year all people take vacations, but all at the same time?)

We did not find examples for the complete example in the Internet. Using asterisk in Google search and a reduced context we obtained 36 pages in the Spanish language search of “*con \* motivo \* de fin de año*”, but only three different types:

1. The prepositional phrase appears without modification (NOTHING)
2. The prepositional phrase is broken and parts of it correspond to different parts of a long phrase (PHRASE)
3. The noun of the prepositional phrase is modified (MODIF)

In order to obtain more examples we reduced the context without changing the prepositional phrase meaning. We limited the phrase to “*con \* motivo \* de fin*”, and we obtained 55 examples. In the following table we present the three different observed types, an example for each type and the total number of obtained items:

NOTHING	... que se celebran en fechas connotadas, generalmente relacionadas <b>con celebraciones históricas o con motivo de las fiestas de fin</b> de año. ... ‘that are celebrated on the corresponding dates, usually related <b>with historical celebrations or on the occasion of the holidays of end</b> of year ...’	34
MODIF	<b>Con el motivo adicional de fin</b> de año, esperamos desde ya su asistencia y participación Jorge Castro, Jorge Raventos, Pascual Albanese 27/11/2006 ... ‘On the additional occasion of the end of year, we expect from right now his presence and participation Jorge Castro, Jorge Raventos , Pascual Albanese 27/11/2006 ...’	4
PHRASE	... Orizaba, Tampico, etcétera, etcétera, han estado cambiando, <b>con este último motivo, telegramas para ponerse de acuerdo a fin</b> de llevar una manifestación ... ‘Orizaba, Tampico, etc., etc. has been changing, on the occasion of the latter, telegrams to agree in order to organize a manifestation ...’	17

We observed that very few snippets for restricted noun modification were obtained for the example when the prepositional phrase corresponded to a  $EXP_{PNP}$  phrase and we expect that much more snippets should be obtained when the prepositional phrase corresponds to a  $REG_{PNP}$ .

Considering the regular phrase *motivo de fin de año* ‘of end of year’, we made a Google search of the string “*de \* fin de año*”. The search gave 651,000 examples. We intended to determine frequencies of the different types of examples obtained. Because of the volume, we cannot determine them thoroughly. To evaluate the true portion of the different types of examples in the automatically counted amounts, we looked through the first hundred snippets and manually analyzing their syntax to prove the type.

In 100 snippets, we manually found three different types: eight snippets for MODIF, 89 for PHRASE and 3 snippets for NOTHING. We expected much more cases of MODIF type, but we found less cases of the prepositional phrase as it. It seems that the relations are inverted between MODIF and NOTHING.

## 4.2 Non-substitutable Nouns

Considering that the noun phrase inside the  $EXP_{PNP}$  cannot be replaced by a synonym, we tried to find how noun phrase could be substitutable inserting its synonyms from a dictionary and searching the modified phrase in the Internet.

The initial phrase “*que con motivo de fin de año*” obtained 220 pages in Google search. We kept the same context and for the phrase “*que con motivo de SINONYM*”

*de año*” we did not obtain results for any synonym. However, reducing the context we obtained:

---

“ <i>con motivo de SINONYM de año</i> ”:	<i>final</i> – 2 pages, <i>terminación</i> – 3 pages
“ <i>motivo de SINONYM de año</i> ”:	<i>final</i> – 2 pages, <i>terminación</i> – 3 pages
“ <i>de SINONYM de año</i> ”:	<i>final</i> – 157000 pages, <i>cabo</i> – 121 pages <i>remate</i> – 12 pages, <i>terminación</i> – 4 pages
	<i>propósito</i> – 132 pages, <i>meta</i> – 4 pages

---

We could observe that for  $REG_{PNP}$  we found many combinations and we did not expect to obtain results for the  $EXP_{PNP}$ .

Considering the same phrase, we searched “*que con SYNONYM de fin de año*” and we did not find results using the synonyms of *motivo* (*causa, razón, pretexto*, etc.). We reduced the context and we only found one page for “*con pretexto de fin de año*” and two pages for “*con pretexto de fin*”.

### 4.3 Part of Idioms

As it is reported in [4], idiomatic expressions combine with a more restricted number of neighboring words than free combinations. They computed three indices on the basis of three-fold hypothesis: a) idiomatic expressions should have few neighbors, b) idiomatic expressions should demonstrate low semantic proximity between the words composing them, and c) idiomatic expressions should demonstrate low semantic proximity between the expression and the preceding and subsequent segments

We consider the first hypothesis since the others hypothesis imply semantic measures. For example, the P–NP–P *a fin de* has more neighbors than the phrase *a fin de cuentas*. Searching for “*a fin de \**” we obtained 1,340,000 pages, manually we observed that a noun was its immediate neighbor and that the noun was linked to the P–NP–P in most of the cases. We obtained 1,220,000 pages for *a fin de cuentas* and manually we observed that the immediate neighbor was not linked to the phrase in most of the cases.

## 5 Association Measures

Diverse statistical measures have been used to identify lexical associations between words from corpus [3, 7]. To formally define and to numerically test the relation between words we did not take the criterion based only on frequencies, since these frequencies depend on the corpus size.

The results that we present in the following subsections were obtained with a collection of examples obtained from 1000 sentences that have one  $EXP_{PNP}$  according to [10]. For each sentence, we extracted each  $P_1$ –NP– $P_2$  with its context.

## 5.1 Noun Modification

For each  $P_1$ -NP- $P_2$  we got the  $P_1 * NP * P_2 + context$  search where asterisk could represent punctuation signs, one word or several words. We wrote a PERL program to determine the type of the obtained string, instead of the asterisk, by means of POS and very simple rules. We classified such type as: ‘modif’, ‘phrase’, ‘nothing’ and ‘doubt’. Doubt means that our POS tool and the heuristics could not assign a type to that string. Considering 100 snippets, we automatically calculate the true portion of each type.

We obtained the results for five hundred examples considering only the MODIF and PHRASE types. We could observe that several of them had the same values since the delimitation of context made equal phrases. Nevertheless, we could observe that a delimitation could be defined for MODIF from zero to 0.22% value.

We did not obtain clearly differentiated values for MODIF and PHRASE types in one thousand examples of  $REG_{PNP}$ .

## 5.2 Substitutable Nouns

The measure that we want to obtain to determine if it is possible to substitute the noun inside the  $P_1$ -NP- $P_2$  is how many combinations exist, i.e., how many of the synonyms are combinable with  $P_1$  and  $P_2 + context$ .

We wrote a program that use a Spanish Synonym Dictionary to substitute each one of their synonyms instead of the NP, then it searches the Internet and it outputs the number of synonyms that obtained a result. Some examples are listed in the following table where we show: the original phrase with the  $P_1$ -NP- $P_2$  between hyphens, an example of the reduced context, the number of synonyms that obtained a result and the number of pages.

Original phrase	Example	#SYN	Pages
propuesta - de acuerdo con - sus intereses	de compromiso con sus intereses	1	4
fin - de llegar a - una posible reglamentación al respecto	de aproximarse a una posible	2	17
calidad - de vida de - sus habitantes	de ocupación de sus habitantes	11	159
fin - de trabajar de - manera coordinada y enfrentar cualquier acto de	fin de luchar de manera	46	1685

The values obtained for the prepositional phrases that obtained results substituting the noun by a synonym are shown in Figure 1. We eliminated such sentences that had a NP without an entry in the Synonym Dictionary, phrases that obtain zero results in the Web search and phrases without context. We could observe that  $REG_{PNP}$  have values between 1 and 25. We could observe that  $EXP_{PNP}$  have less number of combinations. We can neglect the peaks since they correspond to a P-NP-P that is part of an idiom for example: “*a fin de cuentas*” and “*a fin de que*”. We could determine that  $REG_{PNP}$  have differentiated values between 4 and 25 combinations, where  $EXP_{PNP}$  have near zero examples.

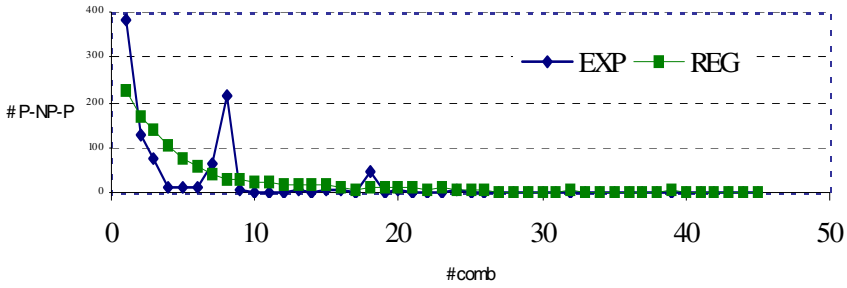


Fig. 1. Number of combinations obtained for the substitution of NP by its synonyms

### 5.3 Lexical Association for Idioms

We use similar criteria of Mutual Information, the Stable Connection Index [2] in the shape

$$SCI(P_1, P_2) \equiv 16 + \log_2 \frac{N(P_1, P_2)}{\sqrt{N(P_1) \times N(P_2)}}$$

where the constant 16 and the logarithmic base 2 are chosen in a way that the majority of the results are in the interval 0 to 16. To calculate *SCI*, we do not need to know the steadily increasing total volume under the search engine’s control. *SCI* reaches its maximally possible value 16 when  $P_1$  and  $P_2$  always go together. *SCI* retains its value when  $N(P_1)$ ,  $N(P_2)$ , and  $N(P_1, P_2)$  change proportionally. This is important since all measured values fluctuate quasi-synchronously in time.

We obtained the *SCI* values for pairs of word groups using the Google search engine considering that  $P_1$  is P–NP–P and  $P_2$  is the right context to evaluate the possibility of P–NP–P being the initial part of an idiom.

For example, we took 219 sentences of newspaper texts where the P–NP–P *al pie de* occurs. We extract each P–NP–P with its context and their *SCI* value was obtained. The range for *SCI* values goes from –2.84 to +13.13. Thirteen phrases have negative values; 35 phrases have zero *SCI* value since the string of words  $P_1 P_2$  got zero occurrences; 9 phrases have *SCI* values between 12.11 and 14.33. All the rest have *SCI* values between 0.16 and 10.8.

We show some examples in the following table:

Phrase	#Pages	Context	# Pages	SCI
al pie de la sede	31	la sede	9900000	-1.44
al pie de la Suburban	0	la Suburban	940	0
al pie de un cactus	9	un cactus	51000	0.57
al pie de una cruz	205	una cruz	964000	2.96
al pie de unas colinas	90	unas colinas	9930	5.07
al pie de la torre	13500	la torre	7210000	7.55
al pie del cerro	52600	cerro	6460000	9.59
al pie de la montaña	49500	la montaña	4480000	10.06
al pie del cañón	140000	cañón	2290000	12.98
al pie de la letra	767000	la letra	10200000	14.33

The group with values greater than 12.98 correspond to idioms, for example: *al pie del cañón, al pie de la letra*. Other five cases showed that idioms obtained SCI values greater than 12.5

We also obtained the Google statistics for the phrases: NP and P<sub>2</sub> + context, to prove the combinability of the noun with the second preposition and the right context. It seems that a high value of SCI<sub>NP-P2+context</sub> and much higher than SCI<sub>P-NP-P+context</sub> correspond to a REG<sub>PNP</sub>

#### 5.4 Heuristics for Use Disambiguation of Prepositional Phrases

The heuristics that we deduce from the previous results are:

1. If the number of synonyms is greater or equal to four and lower than 25, it is probable that the P–NP–P is a REG<sub>PNP</sub>.
2. If modifiers detected like adjectives exist and their values are greater than zero and smaller than 0.2 it is probable that the P–NP–P is an EXP<sub>PNP</sub>.
3. If the SCI values for the complete phrase and noun both are greater than 12.5 and very closer it is probable that the P–NP–P is an idiom.
4. If the noun SCI<sub>NP-P2+context</sub> is greater than the SCI of the complete phrase and the values have a difference greater than 3.0 then it is probable that the P–NP–P is a REG<sub>PNP</sub>.

## 6 Results

We extracted sentences of a newspaper corresponding to 22/12/04. For each sentence we obtained the prepositional phrases P–NP–P + context. For each one we obtained the SCI values for the complete group and for the noun and P2 + context. We also obtained the modifier types (PHRASE, MODIF, NOTHING, DOUBT) of the noun with their True Portion and the statistics of synonymous substitution from Google.

After applying the heuristics described in the previous section, we obtained the following results:

Type	Precision	Recall	# detected	# correct	#correct detected
EXP <sub>PNP</sub>	46	-	39	18	18
IDIOM	100	80	4	5	4
REG <sub>PNP</sub>	99	82	99	120	98

where:

Precision is the number of correct P–NP–P detected / # of P–NP–P detected,

Recall is the number of correct P–NP–P detected / # of P–NP–P manually labeled.

The results show the values obtained only for the P–NP–P that got at least one value, since our Synonym dictionary have few thousands of entries, some contexts were not reduced and the Google search engine did not find hits for them.

We found that among the 21 REG<sub>PNP</sub> bad recognized as EXP<sub>PNP</sub> eight of them correspond to cases where the context coincides exactly with a named entity. Other eight

cases correspond to P–NP–P that got two opposite values, for example, the test for noun modification detects an  $EXP_{PNP}$  and the test for non-substitutable noun detects a  $REG_{PNP}$ . When the context is a named entity the previous noun is highly tied to the name, for example “*el penal de La Loma*” (La Loma prison) so few modifiers and synonyms are used. If we include a heuristic regarding named entities, precision for  $EXP_{PNP}$  increases to 56%. For  $REG_{PNP}$  there was only one case where the P–NP–P got two opposite values. When only two opposite values are obtained the heuristics assign first the value for the noun modification test.

Based on these results, future work will consider to apply a classification method instead of heuristics but more examples will be needed for training them. The very simple rules to determine the type of the retrieved strings of noun modification should be changed to improve the “*modif*” and “*phrase*” recognition.

In addition, a refinement is necessary in the delimitation of context, since one word could be the difference to obtain results from Google. Having results from the search engine the next improvement should deal with inclusion of a bigger quantity of snippets to have a more accurate determination of modifier types.

## 7 Conclusions

Idiomatic word combinations of  $EXP_{PNP}$  type, usually functioning as compound prepositions, have linguistic properties distinct from those regular. In particular, they combine with a greater number of words than usual idioms. For their unsupervised determination from Internet, we proposed to obtain their combinability through Google searching. We presented an approach to differentiate them from fixed phrases and free combinations by means of some association measures according to linguistic properties.

We used as association measures the combinations with synonyms of the noun to prove non-substitutable nouns, the noun modifications by means of adjectives to prove restricted modification and a like-mutual information to deduce the strength of links between the prepositional phrase and the context, and between the noun and the right prepositional phrase. The values obtained for such measures in a test collection were adapted to some heuristics.

The preliminary results obtained from the application of heuristics give some directions to improve their adequacy to determine the degree to which the prepositional phrases are compositional.

## References

1. Baldwin, T., Bannard, C., Tanaka, T., Widdows, D.: An Empirical Model of Multiword Expression Decomposability. In: Dignum, F.P.M. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 89–96. Springer, Heidelberg (2004)
2. Bolshakov, I.A., Galicia-Haro, S.N.: Detection and Correction of Malapropisms in Spanish by means of Internet Search. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 115–122. Springer, Heidelberg (2005)

3. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. In: Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics, pp. 76–83 (1989)
4. Degand, L., Yves, B.: Towards automatic retrieval of idioms in French newspaper corpora. *Literary and Linguistic Computing* 18(3), 249–259 (2003)
5. Dekang, L.: Automatic identification of noncompositional phrases. In: Proc. of the 37th Annual Meeting of the ACL, College Park, USA, pp. 317–324 (1999)
6. de María Moliner, D.: *Diccionario de Uso del Español*. Primera edición versión electrónica (CD-ROM) Editorial Gredos, S. A. (1996)
7. Evert, S., Krenn, B.: Methods for the Qualitative Evaluation of Lexical Association. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics. Toulouse, France, pp. 188–195 (2001)
8. Galicia-Haro, S.N., Gelbukh, A.: Towards the Automatic Learning of Idiomatic Prepositional Phrases. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) *MICAI 2005*. LNCS (LNAI), vol. 3789, pp. 780–789. Springer, Heidelberg (2005)
9. Manning, C.D., Schütze, H.: *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge (1999)
10. Nañez Fernández, E.: *Diccionario de construcciones sintácticas del español*. Preposiciones. Madrid, España, Editorial de la Universidad Autónoma de Madrid (1995)
11. Real Academia Española.: *Diccionario de la Real Academia Española*, 21 edición (CD-ROM), Espasa, Calpe (1995)
12. Seco, M.: *Gramática esencial del español, introducción al estudio de la lengua*, Segunda edición revisada y aumentada, Madrid, Espasa Calpe (1989)
13. Villada, B., Tiedemann, J.: Identifying idiomatic expressions using automatic word-alignment. In: Proceedings of the EACL 2006 Workshop on Multiword Expressions in a Multilingual Context, Trento, Italy (2006)



# Identification of Chinese Verb Nominalization Using Support Vector Machine

Jinglei Zhao, Changxiong Chen, Hui Liu, and Ruzhan Lu

Department of Computer Science  
Shanghai Jiao Tong University  
800 Dongchuan Road Shanghai, China  
{zhaojl, cxchen, lh\_charles, rzlu}@cs.sjtu.edu.cn

**Abstract.** Nominalization is a highly productive phenomenon across languages. The process of nominalization transforms a verb predicate to a referential expression. Identification of nominalizations presents a big challenge to Chinese language processing because there is no morphological difference between a verb nominalization and its corresponding verb predicate. In this paper, we apply a support vector machine to identify nominalizations from text. We explore extensively the various nominalization specific classification features for the identification task. Among which, many are first introduced in the literature. The experimental result shows that our method is very effective.

## 1 Introduction

Verb Nominalization is a common phenomenon across languages where verb predicates are transformed to function as referential objects just as nominal expressions, e.g. *construction* transformed from *construct* in English. Unlike English, verb nominalizations in Chinese have the same form with their corresponding verb predicates. This can be illustrated by the following instances in which the same verb 下棋(play-chess) is a predicate in sentence 1), while a referential expression in 2).

- 1). 他经常下棋 (He often *plays chess*).
- 2). 下棋是很好的活动 (*Playing chess* is a very good activity).

The identification of verb nominalizations is very important to natural language processing (NLP). First, nominalizations are very productive in Chinese, with a percent of 23.5% of the overall verbal occurrences in a balanced corpus [1]. Second, the most essential nature of a language expression is whether it is predicative or referential, which leads to a total difference in syntactic and semantic level representations. The improvement on the identification of verb nominalizations will have a direct influence on many tasks in natural language processing.

Verb nominalizations have some specific grammatical characteristics that provide the basis for its identification. On one side, the semantics of the verb predicate and much of its syntactic structure can be retained by nominalization.

For example, it retains the argument structure. On the other side, nominalizations can enter into the subject or object positions of a verb predicate. Also, nominalizations can combine with nouns to form nominal compounds.

A support vector machine (SVM) is applied for the identification of nominalizations from text. SVM is a supervised machine learning technique which has been shown to perform well in multiple areas of natural language processing including text categorization [2], parsing [3], named entity recognition [4] etc. The key element for applying SVM is to find the most appropriate classification features of the classified object. In this paper, we explore extensively the various possible classification features of verb nominalizations, among which, many features such as Verb Compounding Ability, Noun Compounding Ability etc. are first introduced in the literature of NLP. We believe that the investigation of such features will also benefit other machine learning based tasks in Chinese language processing.

The remainder of the paper is organized as follows: Section 2 describes related works. Section 3 gives the definition of verb nominalization and clarifies the identification problem. Section 4 introduces the support vector machine used for the identification task. Section 5 gives a description of a baseline system for evaluating possible classification features. Section 6 evaluates the performance of the the set of various features and gives the best result system. Section 7 compares the performance of SVM with other two statistical models for the identification task. Finally, in Section 8, we give the conclusions.

## 2 Related Works

Several works have been done in the identification of verb nominalizations. In English, nominalizations have affixes such as -ion, -ment, etc. compared with their related verb predicates. So, the identification task can be easily done by constructing a lexical dictionary listing the possible nominalizations. Nomlex [5] is the most widely used dictionary that lists nominalizations, their related verb predicates, and correspondence between the verbal arguments and syntactic positions within the noun phrase. [6, 7] used Nomlex to recognize nominalizations in question answering (QA) systems. [8–10] identified nominalizations for the purpose of compound interpretation. [11] recognized nominalizations using Nomlex in task of semantic role labeling. Also, by exploiting nominalizations in Nomlex, [12] extracted nominal mentions of events from the text.

Nominalizations in Chinese don't have any lexical inflections compared with the corresponding verb predicates and thus makes the identification task more difficult than that in English. Influenced by the linguistic work of [13], most computational works include nominalization identification in Chinese into a part-of-speech (POS) tagger by considering nominalizations as a subcategory of verbs. [1] is a balanced corpus which distinguishes verb predicates from verb nominalizations in the annotation. [14–16] applied the labeling of verb nominalization from corpus for Chinese base noun phrase (base-NP) recognition. However, no

work has ever investigated or focused on the special problems involved in the identification of nominalizations in Chinese.

### 3 Problem Description

Due to the dubious status of verb nominalization in Chinese linguistic theory, we give the definition of verb nominalization in formal semantics and clarify the identification problem in this section.

First, following Montague [17], a type system *type* for natural language semantics is defined as follows:

- 3) a: *e* is a *type*.
- b: *t* is a *type*.
- c: If *a* and *b* are any types, then  $\langle a, b \rangle$  is a *type*.
- d: Nothing else is a *type*.

The model-theoretic domain  $D_e, D_{\langle e,t \rangle}$  of referential type *e* and predicative type  $\langle e, t \rangle$  is the set of entities and the set of properties with the corresponding syntactic categories  $C_e$  and  $C_{\langle e,t \rangle}$ . Then nominalization can be defined as a type-shifting operator [18, 19]:

$$4) \text{ nom}^{(\cap)} : D_{\langle e,t \rangle} \rightarrow D_e.$$

As a result, if the semantic type of a syntactic expression *P* is of type  $\langle e, t \rangle$ ,  $\cap P$  will be of type *e*.

Given a set of verb occurrences  $\{\langle \text{verb}_i, \text{cont}_i \rangle \mid 1 \leq i \leq N\}$ , where  $\text{cont}_i$  is the corresponding context of  $\text{verb}_i$ , the task of nominalization identification is to identify the possible operator  $\text{nom}^{(\cap)}$  from the contexts of the corresponding verb occurrences.

### 4 Support Vector Machine

A support vector machine is applied for the task of recognizing verb nominalizations from verb occurrences. SVM is introduced by Vapnik [20] based on the maximum margin strategy. Suppose we are given *n* training examples  $(x_i, y_i)$ ,  $(1 \leq i \leq n)$ , where  $x_i$  is a feature vector in *m* dimensional feature space,  $y_i$  is the class label  $\{+1, -1\}$  (positive or negative) of  $x_i$ . SVMs find a hyperplane  $w \cdot x + b = 0$  which correctly separates training examples and has maximum margin which is the distance between two hyperplanes  $w \cdot x + b \geq 1$  and  $w \cdot x + b \leq -1$ .

The optimal hyperplane with maximum margin can be obtained by modeling it as a quadratic programming problem below:

$$\begin{aligned}
 \min \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i, \\
 \text{s.t.} \quad & y_i(w \cdot x_i + b) \geq 1 - \xi_i, \\
 & \xi_i \geq 0
 \end{aligned} \tag{1}$$

where  $C$  is a penalty parameter on the training error,  $\xi_i$  is called a slack variable for a non-separable case. Finally, the optimal hyperplane can be written as:

$$f(x) = \text{sign} \left( \sum_i^n \alpha_i y_i K(x_i, x) + b \right) \quad (2)$$

where  $\alpha_i$  is the Lagrange multiplier corresponding to each constraint of the quadratic programming problem, and  $K(x_i, x)$  is called a kernel function that calculates similarity between two arguments  $x_i$  and  $x$ . SVMs estimate the label of an unknown example  $x$  whether  $\text{sign}$  of  $f(x)$  is positive or not. Specified to our problem, it estimates whether a verb occurrence is a verb predicate or a verb nominalization.

For our experiments, we use LIBSVM [21] as the SVM training and testing software. The system uses a linear kernel and set tolerance of the termination criterion  $e=0.001$ .

## 5 Baseline System

The corpus we use for the identification task is the People's Daily Corpus [1], a one million word lexical annotated corpus of Mandarin Chinese containing one month's data from People's Daily (January 1998). From the annotated corpus, we randomly select 1MB texts that contain 27209 verb occurrences among which 5169 occurrences are verb nominalizations. The above data are randomly partitioned into a training and a testing set containing 21768 and 5441 instances respectively.

To evaluate the effect of various possible classification features on nominalization identification, first, we construct a system which uses only the context words' POS of the verb occurrence as a baseline. The motivation is that, in many tasks of natural language processing, acceptable performance can be got using such features alone. However, the problem lies in how long should the length of the context be. Figure 1 illustrates the influence of the context length on the result of the baseline system. In which, for example, if the context length is 2, it means that both the POS of the two words before and the POS of the two words behind the current verb are used as the classification feature for support vector machine.

The four measures *Precision* (Prec), *Recall* (Rec), *F* and *Rate* for evaluating the system's performance are defined as follows:

$$\begin{aligned} \text{Precision} &= \frac{\text{Number of Correct Identified Nominalizations}}{\text{Number of Identified Nominalizations}} \\ \text{Recall} &= \frac{\text{Number of Correct Identified Nominalizations}}{\text{Number of Nominalizations in Text}} \\ F &= \frac{\text{Precision} * \text{Recall} * 2}{\text{Precision} + \text{Recall}} \\ \text{Rate} &= \frac{\text{Number of Correct Classified Verb Occurrences}}{\text{Number of Verb Occurrences}} \end{aligned} \quad (3)$$

From figure 1, we can see that increasing context length has a consequence of increasing *Recall* but lowering *Precision*. The *F* measure attains the highest

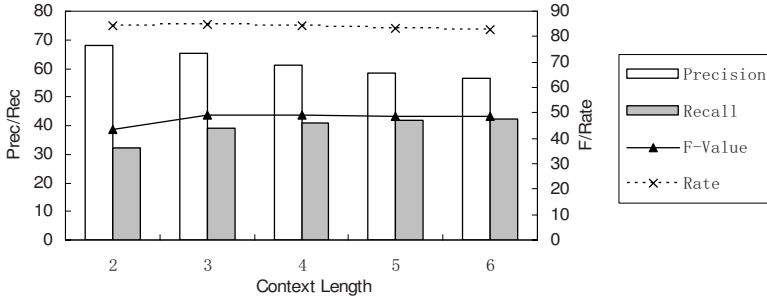


Fig. 1. The effect of context length on the performance of baseline system

value at length 3, while *Rate* doesn't change much respect to the changing of context length. Overall, we select the context length 3 for further evaluation, at which the *F* measure is 49%.

## 6 System Improvements

### 6.1 Features Evaluated

Verb Nominalizations have many specialized characteristics with respect to verb predicates. In this section, we explore extensively the various possible classification features we can use for the identification task.

**Function Words of Context.** Some function words such as prepositions and auxiliary words in the context of a verb can be very good indicators of verb predicates or verb nominalizations. For example, if the word before the verb occurrence is 的 ('s), in most cases, the verb occurrence would be a verb nominalization. Likewise, if the word behind the verb is among aspect markers such as 了 (*le*, indicating a past aspect), 着 (*zhe*, indicating a present aspect) etc, it provides a strong clue that the verb occurrence is a verb predicate. Similarly, tense markers such as 已经 (already), 经常 (often) in the context also serve as negative indicators of verb nominalizations.

**POS Occurrence.** This feature indicates whether a specific POS ever occurred in the context before or after the verb occurrence. Different from the baseline feature, which considers a rigid POS sequence of the verb's context, this feature doesn't consider the position where the POS occurred. Thus, this feature tries to find the most relevance POS for the identification task but ignores others.

**Function Word Occurrence.** This feature is similar to the feature POS Occurrence. The difference is that it considers the most relevant function words, not its occurrence positions in the context.

**Position of Verb Occurrence.** The occurrence order of the verb among all verb occurrences in a sentence can provide some distinction between predicate and nominalization in a statistical sense. For example, over 62% of the first

verb occurrences are the main verbs of the corresponding sentences (estimated from a tree bank of 8000 sentences). Main verb is the outmost predicate in the predicate-argument structure of a sentence which is certainly a verb predicate.

**Verb Compounding Ability.** A large part of verb nominalizations in text combine with other nouns to form nominal compounds, a kind of Multi-Word Expressions (MWE), e.g. 栽培技术 (*planting technology*). The statistic Verb Compounding Ability (VCA) characterizes the capability of a verb to involve in the construction of nominal compounds. Such a statistic is estimated from the Chinese Classification Subject Thesaurus [22], which can be seen as a corpus of compounds, as defined below:

$$VCA(v) = \frac{\text{Occurrences of } v \text{ in Compounds}}{\text{Occurrences of All Verbs in Thesaurus}} \quad (4)$$

**Noun Compounding Ability.** When compounding with nouns, verb nominalizations have preferences for the head nouns. Inversely, the semantic categories of the nouns in context have influences on whether the verb occurrences are nominalizations. For example, verb occurrences before attribute nouns such as 面积 (*area*) and 速度 (*speed*) are more likely to be nominalizations compared to those before the entity denoting nouns such as 苹果 (*apple*). The Noun Compounding Ability (NCA) characterizes the possibility of a noun to be the head noun of nominal compound. Such a statistic is also estimated using the Chinese Classification Subject Thesaurus as (5). If a noun is appeared in the context of the verb occurrence, its NCA score is exploited as the classification features for the verb occurrence.

$$NCA(n) = \frac{\text{Occurrences of } n \text{ as Head}}{\text{Occurrences of All Nouns in Thesaurus}} \quad (5)$$

**Mutual Information.** Nominal compounds are a kind of collocations in some sense. If a noun and a verb co-occur frequently, it may provide some information of the verb occurrence to be a verb nominalization. Mutual information  $MI(x, y)$  is a commonly used criterion for the correlation between the two word  $x$  and  $y$ , with its definition below:

$$MI(x, y) = \frac{f(x, y)}{f(x) * f(y)} \quad (6)$$

**Length of the Verb.** [23] points out that prosodic structures have direct constraints on syntax of Chinese. The number of syllables in a verb occurrence could be a useful indicator differentiating whether the structure it appears is a compound or verbal phrase. For example, the prosodic structure  $V1 + N2$  seldom forms nominal compounds but usually verbal phrases. In the above structure, for instance,  $V1$  means verbs with only one syllable. Motivated by such linguistic theory, the length of the verb occurrence is applied as a differentiating feature between predicates and nominalizations.

**Morphemes of Verb.** Except the length of the verb, the word formation information also provides lexical clues for its syntactic behavior. For example,

if the second morpheme of a verb is 成(-ed), as the verbs 造成(caused), 形成(formed) etc., it will seldom be used as a nominalization.

**Predicate Frequency.** For the nominalization task, the prior probability of a verb to be used as a nominalization maybe the most salient feature. This is estimated by the frequency of a verb occurrence to be a verb predicate, computed from the whole People's Daily Corpus.

**Context Predicate Frequency.** If two verbs appear adjacently or nearly, they will have direct influence on each other for its syntactic property. For example, in coordination phrase 小麦的栽培和收割(*the planting and reaping of wheat*), if the verb occurrence 栽培(*planting*) is very likely to be a verb nominalization, then together with the information of functional word 和(*and*), it will provide validating evidence for 收割(*reaping*) to be a nominalization.

**Support Verb Ahead.** When verbs are deverbalized, semantically empty verbs called support verbs are often employed before nominalizations to achieve syntactic appropriateness. For examples, the support verb 进行(*conduct*) in the structure 进行规划(*conduct plan*). So, support verbs are good nominalization indicators for its following verb occurrences. In some cases, the distance of such a dependence in a sentence is relatively long. For instance, in the sentence below, the distance between the support verb 进行(*conduct*) and the nominalization 预报(*forecasting*) is 4.

- 5). 这台仪器可以进行长期海洋天气预报 (This instrument can *conduct* long-term marine weather *forecasting*).

All the features listed above form feature templates that can product numerous feature instances for the support vector machine.

## 6.2 Feature Performance

Table 1 shows the effect each feature explained in the above section has on the verb nominalization identification task. In which, "Baseline" means using the context tag alone as discussed in section 5, while "+Verb Compounding Ability", for example, means using classification features Verb Compounding Ability plus the baseline feature.

For the four performance measures,  $F$  is the most indicative measure for the overall performance of the system. As expected, the prior probability for a verb to be used as a nominalization has the most influence on the task when adding such feature alone. In such a case,  $F$  measure reaches 73.0%, 24.0 percent above the baseline. However, the feature Mutual Information has very little positive effect on the identification result which is out of our expectation. Overall, the recall of the identification is relatively lower than the precision. In the baseline system the recall is only 39.1% percent. So, improvements on recall will have more impact compared to precision. As we see, many novel verb specified features we proposed, such as verb compounding ability, length of verb etc. improve much on the recall of the identification.

**Table 1.** Effect of the features on the verb nominalization identification task when added to the baseline system(%)

Features	Rate	Precision	Recall	F-Value
Baseline	84.7	65.5	39.1	49.0
+Function words of context	87.0	71.9	50.3	59.2(+10.2)
+POS occurrence	84.7	62.2	46.0	52.9(+3.9)
+Function word occurrence	85.7	66.8	43.4	52.1(+3.1)
+Position of verb occurrence	84.9	65.2	41.6	50.8(+1.8)
+Verb compounding ability	87.3	71.7	53.3	61.1(+12.1)
+Noun compounding ability	85.0	67.8	39.8	50.2(+1.2)
+Mutual Information	84.7	65.4	39.3	49.1(+0.1)
+Length of the verb	85.4	66.1	45.3	53.8(+4.8)
+Morphemes of verb	85.1	65.9	41.7	51.1(+2.1)
+Predicate frequency	90.7	80.0	67.1	73.0(+24.0)
+Context predicate frequency	85.0	66.6	39.7	49.8(+0.8)
+Support verb ahead	85.2	65.9	41.1	50.6(+1.6)

### 6.3 Feature Selection

Given the improvements by various features in the above section, it seems that we can get the best performance system by simply combine all the features with positive effect. However, there are still some problems for the overall system. First, many features are highly correlated, for example, the feature Verb Compounding Ability are indubitably correlated with the feature Predicate frequency. When combining together, features may have negative influences on each other. Second, some feature templates like Functional Words of Context can form large amount feature instances. As an illustration, if the scale of the lexicon of functional words is 2,000 and the context length is 3, it will form 12,000 feature instances. Among which, many feature instances are irrelevant to our task or even act as noises in the entire feature set for the support vector machine. Finally, one shortcoming of SVM is that the model's predicting time is very long. So, for practical considerations, it is very important for the model to cut off the irrelevant feature instances to speed the generalization process.

Feature selection can reduce the dimensionality of the input space and improve the generalization error [24]. We use an feature selection criterion *F-score* [25] to get the best performance system with relatively small input space.

*F-score* is a simple technique which measures the discrimination of two sets of real numbers. Given training vectors  $x_k, k = 1, \dots, n$ , if the number of positive and negative instances are  $n_+$  and  $n_-$ , respectively, then the *FScore* of the  $i^{th}$  feature is defined as:

$$F(i) = \frac{\left(\bar{x}_i^{(+)} - \bar{x}_i\right)^2 + \left(\bar{x}_i^{(-)} - \bar{x}_i\right)^2}{\frac{1}{n_+ - 1} \sum_{k=1}^{n_+} \left(x_{k,i}^{(+)} - \bar{x}_i^{(+)}\right)^2 + \frac{1}{n_- - 1} \sum_{k=1}^{n_-} \left(x_{k,i}^{(-)} - \bar{x}_i^{(-)}\right)^2} \quad (7)$$



where  $\bar{x}_i$ ,  $\bar{x}_i^{(+)}$ ,  $\bar{x}_i^{(-)}$ , are the average of the  $i^{th}$  feature of the whole, positive, and negative data sets, respectively;  $\bar{x}_{k,i}^{(+)}$  is the  $i^{th}$  feature of the  $k^{th}$  positive instance, and  $\bar{x}_{k,i}^{(-)}$  is the  $i^{th}$  feature of the  $k^{th}$  negative instance. The numerator indicates the discrimination between the positive and negative sets, and the denominator indicates the one within each of the two sets. The larger the *F-score* is, the more likely this feature is more discriminative between positive and negative instances. Specified to our problem, the higher *F-score* means the feature can provide more discriminative power between verb predicates and verb nominalizations.

After computing the *F-score* of all the features produced by the various feature templates listed in the last section. Features with a *F-score* less than a threshold  $N$  (0.0001 in the experiment) is discarded from the feature set. Table 2 gives the result of the overall system. By combining all the features with positive effect in Table 1, *F* measure achieves 84.1% percent. After feature selection, a comparable performance,  $F = 84.3\%$ , is achieved but with much smaller input space.

**Table 2.** Best Performance System

Features	Rate	Precision	Recall	F-Value	Input-Space
ALL	93.3	86.2	82.1	84.1	27572
After Feature Selection	93.2	85.9	82.8	84.3	4374

## 7 Comparing Classifiers

Finally, in our experiment, two other most popular models in natural language processing are compared to SVM for the verb nominalization identification task. One is hidden markov model (HMM) [26], the other is maximum entropy model (ME) [27].

When testing HMM, we put the verb nominalization identification task under a more general POS tagging background by viewing verb nominalization as a syntactic sub-category of verbs. HMM mainly uses the state transition probability and state emission probability as the features for the labeling task. When testing ME, however, we put it in the same background as the SVM classifier, that is, using the same feature set we proposed for the SVM classifier in section 6. The comparison result is illustrated in Table 3.

From Table 3, we can see that SVM performs a little better than ME using the features specified for the distinction between verb nominalizations and verb predicates. HMM performs poorly using only common features. This comparison further validates the effectiveness of the feature set we proposed for the task of nominalization identification.

**Table 3.** Comparing SVM with Other Models

Features	Precision	Recall	F-Value
support vector machine	86.2	82.1	84.1
hidden markov model	78.9	76.2	77.5
maximum entropy model	85.5	81.5	83.5

## 8 Conclusions

In this paper, we dealt with the problem of identifying the nominalized use of verbs (verb nominalizations) which had never been thoroughly investigated before in Chinese language processing. We defined verb nominalization as a type-shifting operator and discussed the significance for its recognition. Then, we applied a support vector machine for the identification task. We explored extensively the various possible classification features to differentiate between verb predicates and verb nominalizations. Among which, many are first introduced in the field of natural language processing. The experiment result shows that the set of features we proposed is very effective for the identification task.

## Acknowledgements

This work is supported by NSFC Major Research Program 60496326 :Basic Theory and Core Techniques of Non Canonical Knowledge.

## References

1. Yu, S., Zhu, X.: Guideline for Segmentation and Part-Of-Speech Tagging on Very Large Scale Corpus of Contemporary Chinese. *Journal of Chinese Information Processing* 14(006), 58–64 (2000)
2. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: *Proceedings of the 10th European Conference on Machine Learning*, pp. 137–142 (1998)
3. Kudo, T., Matsumoto, Y.: Chunking with support vector machines. In: *North American Chapter Of The Association For Computational Linguistics*, pp. 1–8 (2001)
4. Isozaki, H., Kazawa, H.: Efficient support vector classifiers for named entity recognition. In: *Proceedings of the 19th international conference on Computational linguistics*, vol. 1, pp. 1–7 (2002)
5. Macleod, C., Grishman, R., Meyers, A., Barrett, L., Reeves, R.: Nomlex: A lexicon of nominalizations. In: *Proceedings of the 8th International Congress of the European Association for Lexicography*, pp. 187–193 (1998)
6. Monz, C., de Rijke, M.: Tequesta: The University of Amsterdam’s textual question answering system. In: *The Tenth Text REtrieval Conference (TREC 2001)*, pp. 519–528 (2001)

7. Schwitter, R., Rinaldi, F., Clematide, S.: The Importance Of How-Questions in Technical Domains. In: Proceedings of the Question-Answering workshop of TALN, vol. 4 (2004)
8. Grover, C., Lascarides, A., Lapata, M.: A comparison of parsing technologies for the biomedical domain. *Natural Language Engineering* 11(01), 27–65 (2005)
9. Lapata, M., Lascarides, A.: A probabilistic account of logical metonymy. *Computational Linguistics* 29(2), 261–315 (2003)
10. Nicholson, J.: Statistical Interpretation of Compound Nouns. PhD thesis, University of Melbourne (2005)
11. Pradhan, S., Sun, H., Ward, W., Martin, J.H., Jurafsky, D.: Parsing Arguments of Nominalizations in English and Chinese. In: Proc. of HLT-NAACL (2004)
12. Creswell, C., Beal, M.J., Chen, J., Cornell, T.L., Nilsson, L., Srihari, R.K., Janya, I.: Automatically Extracting Nominal Mentions of Events with a Bootstrapped Probabilistic Classifier
13. Zhu, D.: Lectures on Grammar. The Commercial Press, Beijing (1982)
14. Zhao, J., Huang, C.: A transform-based model for Chinese base noun phrase recognition. *Journal of Chinese Information Processing* 13(002), 1–7 (1999)
15. Zhao, J., Huang, C.: A Probabilistic Chinese BaseNP Recognition Model Combined with Syntactic Composition Templates. *Journal of Computer Research and Development* 36(011), 1384–1390 (1999)
16. Zhao, T., Yang, M., Liu, F., Yao, J., Yu, H.: Statistics based hybrid approach to Chinese base phrase identification. In: Proceedings of the second workshop on Chinese language processing: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics, vol. 12, pp. 73–77 (2000)
17. Montague, R.: The proper treatment of quantification in ordinary English. *Approaches to Natural Language* 49, 221–242 (1973)
18. Chierchia, G.: Topics in the syntax and semantics of infinitives and gerunds. Garland Pub., New York (1988)
19. Partee, B.: Noun phrase interpretation and type-shifting principles. *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers* 8, 115–143 (1987)
20. Vapnik, V.N., Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (2000)
21. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. *Software* 80, 604–611 (2001), available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
22. Chen, S., et al.: Indexing Manual for the Chinese Classification Subject Thesaurus. Beijing Library Press (1998)
23. Feng, S.: Prosodic structure and prosodically constrained syntax in Chinese. PhD thesis, University of Pennsylvania (1995)
24. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing Multiple Parameters for Support Vector Machines. *Machine Learning* 46(1), 131–159 (2002)
25. Chen, Y.W., Lin, C.J.: Combining SVMs with various feature selection strategies. Department of Computer Science and Information Engineering (2005)
26. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
27. Berger, A.L., Pietra, V.J.D., Pietra, S.A.D.: A maximum entropy approach to natural language processing. *Computational Linguistics* 22(1), 39–71 (1996)

# Enrichment of Automatically Generated Texts Using Metaphor

Raquel Hervás<sup>1</sup>, Rui P. Costa<sup>2</sup>, Hugo Costa<sup>2</sup>, Pablo Gervás<sup>1</sup>,  
and Francisco C. Pereira<sup>2</sup>

<sup>1</sup> Instituto de Tecnologías del Conocimiento  
Universidad Complutense de Madrid, Spain  
raquelhb@fdi.ucm.es, pgervas@sip.ucm.es

<sup>2</sup> CISUC, Department of Informatics Engineering  
University of Coimbra, Portugal

{racosta,hecosta}@student.dei.uc.pt, camara@dei.uc.pt

**Abstract.** Computer-generated texts are yet far from human-generated ones. Along with the limited use of vocabulary and syntactic structures they present, their lack of creativeness and abstraction is what points them as artificial. The use of metaphors and analogies is one of the creative tools used by humans that is difficult to reproduce in a computer. A human writer would not have difficulties to find conceptual relations between the domain he is writing about and his knowledge about other domains in the world, using this information in the text avoiding possible confusion. However, this task is not trivial for a computer. This paper presents an approach to the use of metaphors for referring to concepts in an automatically generated text. From a given mapping between the concepts of two domains we intend to generate metaphors for some concepts relating them with the target metaphoric domain and insert these metaphorical references in a text. We also study the ambiguity induced by metaphor and how to reduce it.

## 1 Introduction

The great challenge for Natural Language Generation (NLG) is known to be one of choice rather than ambiguity. Where natural language understanding has to deal with ambiguity between different possible interpretations of an input, NLG has to decide between different possible ways of saying the same thing. In recent years, natural language generation is slowly considering other domains of application where the choice available for formulating a given concept is much wider. Applications such as the generation of poetry [1] or fairy tales [2] present a wider range of decision points during the generation process than medical diagnosis [3] or weather reports [4].

In domains where the generated texts are more narrative than technical, the differences from the point of view of quality and naturalness between human and computer-generated texts are bigger. Not only the linguistic information used by computers is more restricted, but they also lack creativeness and abstraction

capabilities. Analogy and metaphor mechanisms are some of the creative tools used by humans that are difficult to reproduce in a computer. While the human mind deals perfectly with abstraction and conceptual relations between different domains, a computer must have all this kind of knowledge stored in the form of information and specific heuristics to deal with it.

In the present paper we address a small part of this problem. From available conceptual information of different domains it is possible to find semantic correspondences between the concepts belonging to them. These similarities can be used as a base to produce metaphorical references for some of the concepts, but they are not enough to generate an intelligible metaphor. Both non-common properties of the concepts and context of the discourse must be taken into account when generating a metaphor. We present solutions for the different tasks involved in the process, and study their results and limitations.

## 2 Related Work

Two lines of research are reviewed to provide the basis for understanding the work presented here: structure alignment as means for identifying analogies between domains, and natural language generation technology.

### 2.1 Metaphor and Structure Alignment

It is widely accepted that many of the problems of metaphor interpretation can be handled using established analogical models, such as the structure alignment approach [5]. The general idea behind this approach is that Metaphor fundamentally results from an interaction between two domains (the vehicle and the tenor, in Metaphor literature). This interaction can be simplified as an isomorphic alignment (or mapping) between the concept graphs that represent the two domains. Thus, we see here a domain as being a semantic network (nodes are concepts; arcs are relations), and a mapping between two concepts (of two domains) results from the application of rules that rely on graph structure: if two nodes share the same connection to the same node, they form a potential mapping (triangulation rule [6]); if two nodes share the same connection to other two nodes that are forming a mapping, they form a potential mapping (squaring rule [6]). Since the domain mappings must be isomorphic (1-to-1), there may be many possibilities. Previous attempts at exploring metaphor generation [7] have followed a floodfill probabilistic algorithm based on Divago's Mapper as described in [8]. This alignment algorithm is extremely knowledge-dependent. On the other side, given the complexity of the task (graph isomorphism search), domains too large will become unpractical. To overcome this dilemma, the Mapper is designed not to bring the optimal solution. It uses a probabilistic approach at some choice points, thus potentially yielding different results in each run.

A mapping (say, from a concept X to a concept Y) produced by a structure alignment should emphasize some particular correspondence between two concepts, namely that, according to some perspective, the role that one concept

has on one domain (say, the concept Y in the domain T) can be projected to its counterpart in the other domain (say, the concept X in Z). This also implies the implicit projection of the surrounding context (e.g. its function, properties) directly related the concept X to the concept Y. For example, when someone says “My surgeon is a butcher”, some immediate functions and properties of “surgeon” are projected to “butcher”, such as being “clumsy” (as opposed to “delicate”) or using a “cleaver” (rather than a “scalpel”). These in turn provide other inferences, thus allowing for more elaborate descriptions, such as “clinical slaughterhouse” for “hospital”. These properties of Metaphor become valuable for tasks such as NLG and its potential is clearly large. Algorithms such as those provided by Sapper [6], Mapper [8] or SME [5] help find suitable mappings that are the needed seed for its use. This in itself raises a number of challenges, some of which are being considered in our work.

## 2.2 Natural Language Generation

The general process of text generation [9] takes place in several stages, during which the conceptual input is progressively refined by adding information that will shape the final text. During the initial stages the concepts and messages that will appear in the final content are decided and these messages are organised into a specific order and structure (*content planning*), and particular ways of describing each concept where it appears in the discourse plan are selected (*referring expression generation*). This results in a version of the discourse plan where the contents, the structure of the discourse, and the level of detail of each concept are already fixed. The *lexicalization* stage that follows decides which specific words and phrases should be chosen to express the domain concepts and relations which appear in the messages. A final stage of *surface realization* assembles all the relevant pieces into linguistically and typographically correct text. These tasks can be grouped into three separate sets: *content planning*, *sentence planning*, involving the second two, and *surface realization*.

The appropriate use of referring expressions to compete with human-generated texts involves a certain difficulty. According to Reiter and Dale [9], a referring expression must communicate enough information to identify univocally the intended referent within the context of the current discourse, but always avoiding unnecessary or redundant modifiers. When looking for a reference for a specific concept in the text, it is possible to decide between using a pronoun, the plain name of the concept, its proper noun (if any), a description using its attributes, a description using its relations with other concepts, etc. The range of choice depends directly on the available knowledge.

Reiter and Dale [10] describe a fast algorithm for generating referring expressions in the context of a natural language generation system. Their algorithm relies on the following set of assumptions about the underlying knowledge base that must be used. Every entity is characterised in terms of a collection of attributes and their values. Every entity has as one of its attributes some type. The knowledge base may organise some attribute values as a subsumption hierarchy. For each object, there must also be some way of determining if the

user - the person for which the system is generating text - knows whether a given attribute-value pair applies to it. This serves to determine whether mention of a particular characteristic will be helpful to the user in identifying the object. To construct a reference to a particular entity, the algorithm takes as input a symbol corresponding to the intended referent and a list of symbols corresponding to other entities in focus based the intended referent, known as the *contrast set*. The algorithm returns a list of attribute-value pairs that correspond to the semantic content of the referring expression to be realised. The algorithm operates by iterating over the list of available attributes, looking for one that is known to the user and rules out the largest number of elements of the contrast set that have not already been ruled out.

### 2.3 Metaphor Generation

Little research has been devoted to the generation of metaphors and their use in an automatically generated text. Pereira et al. [7] aimed at improving the stylistic quality of the texts generated by the PRINCE system by extending its capabilities to include the use of simple rhetorical figures. PRINCE (*Prototipo Reutilizable Inteligente para Narración de Cuentos con Emociones*) is a natural language generation application designed to build texts for simple fairy tales. The goal of PRINCE is to tell a story received as input as close as possible to the expressive way in which human storytellers would. To achieve this, PRINCE operates on the conceptual representation of the story, determining what is to be told, how it is organised, how it is phrased, and which emotions correspond to each sentence in the final output. A lexical resource and structure mapping algorithms are used as outlined above to enhance the output texts with simple rhetorical tropes such as simile, metaphor, and analogy.

Pereira et al. identified several problems in the results obtained. From the point of view of interpretation by a reader, metaphorical references and analogies could be confusing if the explicit and implicit information on which they are based is not carefully managed. In addition, once they had identified a metaphor for a specific concept all the appearances of this concept were substituted by the metaphorical reference. As a result the texts were overloaded with analogies, and they presented a significant departure from the original meaning that was difficult to understand. From the point of view of text generation, PRINCE also lacked the linguistic tools required for a correct use of the metaphorical structures. The internal representation of references in PRINCE did not allow the system to refer to any concept using sets of attributes or nominal phrases, so the metaphorical references were reduced to replacing the initial concept with the word assigned to the vehicle concept.

## 3 Generating and Using Metaphors

The process of generating a metaphorical utterance for referring to a specific concept involves several tasks that must be faced separately. Considering that

the initial concept belongs to a given domain, the first step is to find another domain where to look for the desired metaphors. Once it is found, and the mapping between the two domains is established, for each concept susceptible of being referred to using a metaphor a set of possible metaphorical references must be generated. This set of metaphorical references must be studied and evaluated in terms of clearness and suitability, so that inappropriate metaphors can be filtered out. Finally, for each occurrence of the concept in a given context within the text it is necessary to decide whether to use one of the metaphors generated to refer to the concept at that stage or not, always avoiding loss of meaning or unnecessary ambiguity.

The application described in this paper relies on the TAP (*Text Arranging Pipeline*) software architecture for the text generation functionality [11]. TAP is a set of interfaces that define generic functionality for a pipeline of tasks oriented towards natural language generation, from an initial conceptual input to surface realization as a string, with intervening stages of content planning and sentence planning. This process is applied to: the input that is to be processed, the intermediate representations used to store the partial results of progressively filtering, grouping and enriching the input into forms closer and closer to natural language in structure and content, and the set of tasks that take place as steps in that process. The particular instance of the TAP architecture employed here involves three basic modules: a *Content Planner*, a *Sentence Planner*, and a *Surface Realizer*. These modules are organised as a typical basic pipeline for text generation, where the information flows sequentially between the modules that deal with the different tasks involved in the process. Two specific modules have to be considered for the generation and use of metaphors. The Content Planner is the module in charge of deciding which conceptual information from the input would be exposed in the text. It is also in this module where the mapping between domains must be performed, as discussed below. The Reference Solver is a submodule of the Sentence Planner where a referring expression is chosen for each occurrence of a concept in the text. It is in this stage where the metaphorical references must be constructed, and where the decision of whether to use them or not is taken.

### 3.1 Metaphor in the Content Planner: Identifying the Target Domain and the Mapping

The task of selecting target domains and building the mappings has been located within the Content Planner module because this is the part of the generation pipeline that concentrates on handling a purely semantic representation of the data, in the sense that it has not yet been converted into messages susceptible of being relayed in a linguistic form. Therefore, it makes sense to carry out here the mapping operations, which take place over a semantic form of the domains with no reference to their possible linguistic communication. At subsequent stages of the pipeline, the semantic information required will not be available.

The task of identifying an appropriate additional domain as target domain for the metaphor is quite complex. Given that the metaphor is required to contribute



to an act of communication, it is reasonable to say that in order to be appropriate as target domain in an metaphor, a domain must be sufficiently well known to the intended readers of the text so as to require no additional explanation. This narrows down the set of possible domains. It also makes the solution to the problem depend on the particular reader for which the text is intended. Since this requires some means of representing the intended reader as part of the process of generation, for the time being we consider the target domain as given. Further work must focus on exploring the role of reader representation on the choice of target domains.

The generation of each new mapping is firstly dependent on the choice of the pair of domains chosen, the source and the target. In the work here described, we intend to explore the linguistic reference of a concept  $X$  in terms of a target domain  $D$ . For example, how can we reference the concept “excalibur” in terms of the Star Wars saga? To find an answer to this question, we have to use as source the concept map that describes  $X$  (for “excalibur”, it might be a concept map expressing the relations: “excalibur is a weapon”, “excalibur is narrow”, etc.) and as target the domain  $D$  (for Star Wars, it would include relations such as “Han Solo loves Princess Leia” or “Light Saber is a weapon”).

For the current purpose, a Java implementation of the algorithm described in [8] has been developed. In this implementation, known as jMapper, the original algorithm has been slightly modified to improve its efficiency and scalability, although maintaining its general principles. To reduce the search space, the pairs of candidates are ranked in terms of potential similarity. This potential similarity is directly dependent on the number of shared relations that the two concepts have (e.g. dog is more similar to cat than to car, both have legs, breathe, are pets, etc.) and thresholds are established that avoid the exploration of unpromising portions of the search space.

Given the source and the target, the mapping algorithm (jMapper) thus starts looking for initial seeds to start with. This is based on finding pairs of concepts that share the same relation to a third concept (the triangulation rule). The ones that present higher ranks (more shared concepts) will start the process of looking for 1-to-1 correspondences as briefly described above (and in [8]). Several mappings can potentially emerge from this process, but jMapper (unlike previous versions of this algorithm) eliminates most of these during the generation, thus reaching in the end the largest one that could be found from the chosen seed. It is important, however, to notice that this is not an optimal algorithm, as the choice of other seeds could lead to different results. On the other hand, this version is considerably more efficient in terms of computational resources.

### 3.2 Metaphor as Referring Expression Generation During Sentence Planning

In order for the system to be able to use metaphors as references to concepts occurring in the input, it must first construct suitable metaphors, and then it has to decide for which particular occurrences of the concept in the text a metaphorical reference is suitable.

**Constructing Metaphorical References.** The subtask of constructing metaphorical references must be carried out taking the two inputs domains and the mapping between them. This constitutes an additional task not usually contemplated in a natural language generation pipeline.

Once a mapping between two domains has been established, it is necessary to decide which information is useful to generate a metaphorical reference for a specific concept. For each pair of concepts mapped together, a list of their common features that have produced the correspondence is given. Apart from these features, the target concept may have extra attributes not belonging to the vehicle concept. This extra information must be used in the generated metaphor not only to describe the target concept, but also to distinguish it from the vehicle one (conceptually, if two concepts share all their features they are the same concept).

In the representation of the world we are working with, each appearing concept or referent is described by a set of properties. Some examples in two domains are shown in Table 1.

**Table 1.** Examples of properties for two domains

Star Wars domain	Some Properties
storm_trooper	[warrior,man,person,evil]
light_saber	[hand_held,narrow,long,weapon]
princess_leia	[beautiful,young,royal_personage,independent,brunette,...]
King Arthur domain	Some Properties
knight	[warrior,man,person,medieval]
excalibur	[hand_held,narrow,long,weapon,magical,steely,...]
guinnevere	[beautiful,young,royal_personage,queen,blonde,...]

A usual reference for a concept would include some of these properties or attributes to describe it. The idea behind the metaphor is to omit some of these attributes from the reference to this concept by replacing the name of the original concept in the reference with the name of a vehicle concept such that these properties are part of the definition of the vehicle concept. Then the reader will understand as properties of the concept the ones explicitly mentioned along with the ones inferred from the metaphor. For example, the concept “lawyer” may have as attributes to be ‘cunning’ and ‘well-turned out’, and a possible reference to it will be “the cunning and well-turned out lawyer”. If this concept is mapped with the concept “shark”, that is known to be also ‘cunning’, the resulting metaphorical reference would be “the well-turned out shark”.

**Introducing Metaphor in Text.** Metaphor references can not be studied as an isolated phenomenon. In many cases, the context provided by a text is necessary to guide the reader through the assumptions that he must follow to grasp the meaning of the metaphor. But in an appropriate context this metaphor would be understood perfectly. Consider the given example of the lawyer and

the shark where the metaphorical reference “the well-turned out shark” can be hardly understood if the reader does not know we are talking about “lawyers”. However, in the sentence “It was the well-turned out shark who won the trial” the word “trial” submerses the sentence in the legal domain and the metaphor is easily inferred. Once the metaphors have been generated, they become an additional option among all those available as possibilities of referring to that particular element. These possibilities usually include: a pronoun, its proper name, and the name of the class to which the element belongs.

The Referring Expression Generation module is in charge of the task of carrying out the selection of the correct reference of a concept at each and every one of its occurrences in the text. The references to a concept will usually be different at each occurrence, to ensure that the text is fluent and reads naturally. Part of its task includes not only selecting a particular type of reference but also ensuring that the chosen type of reference is appropriately enriched with properties that the concept satisfies so as to ensure that the reference is unambiguous in the context. Within the TAP instantiation employed here, this is carried out by an implementation of the Reiter and Dale algorithm, conveniently enriched to allow for the use of pronouns and proper names, which were not contemplated in the original algorithm.

The task of introducing the metaphorical references in the text deciding where in the text they will be appropriately understood is not exhaustively addressed in the present paper. It is sufficient to say that the algorithm for generating referring expression is extended to include metaphor as an additional option. The TAP reference solver includes heuristics to establish which additional properties of a concept must be mentioned to ensure unambiguous reference when using the name of the class it belongs. These heuristics are used to work out which properties should accompany a metaphorical reference to ensure it is easy to understand. The experiments described below are focused on using the metaphorical references only when all the properties that gave rise to the mapping have already been mentioned in the previous discourse.

## 4 Experiments

In order to test the metaphorical capabilities of our system we have resorted to the use of domain data generated in the past for previous research on Metaphor and Analogy, Tony Veale’s Sapper as reported in his PhD thesis [6]. These data have two distinct advantages. On one hand they constitute a set of coherent domain data already tested for the existence of structural analogies. On the other hand, they were generated independently of the current research effort so they are less likely to be biased towards obtaining interesting results with the proposed method.

Out of the complete data set used in Veale’s thesis, two well known domains have been used: King Arthur saga (target domain) and Star Wars (vehicle domain). The former has been chosen to represent simple referents in our generation system, including the most typical relations of the characters and elements of the domain. The latter is the domain from which metaphors are extracted

to refer to concepts in the first one. Part of the knowledge originally encoded for these domains has been excluded, namely relation weights, and some specific kinds of concepts (compound narrative relations, e.g. `become_arthur_king`, `conceive_morgana_mordred`). Thus, for the moment, we are focussing on the properties of characters, objects and their first order relations within the story (e.g. `have`, `friend_of`, `teach`, `loves`, etc.).

Some of the associations returned as part of a mapping are solely based on very simple general relations such as *gender* or *isa*. Such analogies are considered to be uninteresting and they are discarded. In this example the obtained mapping is shown in Table 2. For each association the list of relations that have produced the mapping and the strength of the analogy is shown.

**Table 2.** Resulting mapping between StarWars and King Arthur domains

Cross domain association	Supporting Relations	Strength
<i>obi_wan_kenobi</i> ↔ <i>merlin</i>	[good,powerful,wise,old,magician,person,man]	0.52
<i>storm_trooper</i> ↔ <i>knight</i>	[warrior,man,person]	0.66
<i>light_saber</i> ↔ <i>excalibur</i>	[hand_held,narrow,long,weapon]	0.73
<i>han_solo</i> ↔ <i>lancelot</i>	[skilful,brave,handsome,young,man,person]	0.43
<i>princess_leia</i> ↔ <i>guinnevere</i>	[beautiful,young,royal_personage,person,woman]	0.63

By following the algorithm explained in Section 3.2, and using the properties and mapping of Tables 1 and 2, the following metaphorical references are produced:

1. “The medieval storm trooper” instead of “knight”. In this case both concepts share many properties, but the attribute ‘medieval’ belonging to ‘knight’ lets us distinguish between the two concepts and facilitates the correct inferences required to understand the metaphor.
2. “The steely light saber” instead of “Excalibur”. As in the previous example, the concepts are distinguished by an extra property of Excalibur, while it also facilitates the readiness of the reference. However, in this case the first concept that comes to mind when reading this reference is ‘sword’ instead of ‘Excalibur’. We will address this issue in the discussion.
3. “Blonde Princess Leia” instead of “Guinnevere”. Here Guinnevere is supposed to be blonde while Princess Leia is a brunette. However, this metaphorical reference is completely unintelligible. This case is also addressed in the discussion.

## 5 Discussion

The generation of metaphors is a delicate process as a metaphor will only be understood if it is possible for the reader to find enough commonalities between the metaphorical concept used and the real concept. For example, in the example we have found a mapping between Lancelot and Han Solo: both of them are skilful

men, brave and young. However, if we generate the sentence “Han Solo arrived to Camelot”, the metaphor will not be understood. This is exactly the case of the third metaphorical reference shown in section 4, concerning Guinnevere and Princess Leïa. Also in the second example it seems that the same problem is partially found when the concept inferred from the metaphor is not “Excalibur” but the class to which it belongs: “sword”.

The explanation for these results is the mixed use of concepts and instances of concepts. Words like “sword” or “knight” are general classes that are defined by a set of properties. We can refer to them as concepts in a general sense. And all the specific individuals that belong to these classes, such as “Excalibur” or “Lancelot” are in fact instances of these concepts. In the experiments performed both concepts and instances were treated in the same way. However, the results have shown that inferring properties from or to a specific instance of a concept is more difficult than doing the same between general concepts. This fact agrees with the theory of Glucksberg and his colleagues [12] who argued that metaphors are interpreted as category-inclusion assertions of the form *X is a Y*. According to this proposal interpreters infer from a metaphor a category (a) to which the topic concept can plausibly belong, and (b) that the vehicle concept exemplifies. When using an instance in the metaphor (as for example, is some of the cases above when a particular person is mentioned by his/her proper name) none of these assumptions is fulfilled.

## 6 Conclusions and Future Work

In this work we have focused in a quite simple vision of what is a metaphor. A valid metaphor is not only supposed to have many features in common with the initial concept, but also to provide extra information belonging to both concepts and not mentioned explicitly for the initial one. In addition, these extra features must be salient in the metaphorical concept used, or the reader may not identify them as also attributed to the target concept [12]. For example, in the sentence “His lawyer is a shark” the metaphor **lawyer** ↔ **shark** is providing implicit information about the lawyer: he is vicious and cunning. This view of metaphor will be explored with more detail in the future.

In our present approach towards the generation of metaphorical references we have only used the properties belonging to concepts and instances when referring to them. However, elements in general domains are also related with other ones by different kinds of relations. For example, in the domains used we can find relations such as “Han Solo is in love with Princess Leïa” or “Excalibur is stuck in a stone”. These relations could not only be used during the generation of the mapping between domains, but also when the metaphorical reference is created as in “the steely light saber stuck in a stone”. This kind of information may make it easier to understand the metaphors. In contrast with the PRINCE fairy tale generator, TAP is capable of realizing into text this kind of linguistic structures, so this issue will be studied in the future.

The requirement of distinguishing between concepts and instances when facing the generation of metaphors suggests that the use of ontologies, where this distinction is explicitly managed, could be a point to study in the future. The characteristic structure of ontologies would not only permit us to differentiate between concepts and instances, it would also allow us to play with their more or less specific properties depending on whether they are properties of a concept or of an instance, or whether they are inherited from more general concepts. With the use of such a taxonomy of properties, new ways of deciding if a metaphor is suitable for a concept may be addressed.

## References

1. Manurung, H.: An evolutionary algorithm approach to poetry generation. PhD thesis, School of Informatics, University of Edinburgh (2003)
2. Callaway, C., Lester, J.: Narrative prose generation. In: Proceedings of the 17th IJCAI, Seattle, WA, pp. 1241–1248 (2001)
3. Cawsey, A., Binsted, K., Jones, R.: Personalised explanations for patient education. In: Proceedings of the 5th European Workshop on Natural Language Generation, pp. 59–74 (1995)
4. Goldberg, E., Driedgar, N., Kittredge, R.: Using natural-language processing to produce weather forecasts. *IEEE Expert* 9, 45–53 (1994)
5. Gentner, D.: Structure-mapping: A theoretical framework for analogy. *Cognitive Science* 7 (1983)
6. Veale, T.: Metaphor, Memory and Meaning: Symbolic and Connectionist Issues in Metaphor Interpretation. PhD Thesis, Dublin City University (1995)
7. Pereira, F., Hervás, R., Gervás, P., Cardoso, A.: A multiagent text generator with simple rhetorical abilities. In: Proceedings of the AAAI-2006 Workshop on Computational Aesthetics: AI Approaches to Beauty and Happiness, AAAI Press, Stanford (2006)
8. Pereira, F.C.: Creativity and AI: A Conceptual Blending Approach. Mouton de Gruyter (2007)
9. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Cambridge University Press, Cambridge (2000)
10. Reiter, E., Dale, R.: A fast algorithm for the generation of referring expressions. In: Proceedings of the 14th conference on Computational linguistics, Association for Computational Linguistics, Morristown, NJ, USA, pp. 232–238 (1992)
11. Gervas, P.: TAP: a text arranging pipeline. Technical report, Natural Interaction based on Language Group, Universidad Complutense de Madrid, Spain (June 2007)
12. Glucksberg, S.: Beyond literal meaning: The psychology of allusion. *Psychological Science* 2, 146–152 (1991)

# An Integrated Reordering Model for Statistical Machine Translation

Wen-Han Chao<sup>1</sup>, Zhou-Jun Li<sup>2</sup>, and Yue-Xin Chen<sup>1</sup>

<sup>1</sup> National Laboratory for Parallel and Distributed Processing, Changsha, China  
{cwh2k, yxchen88}@163.com

<sup>2</sup> School of Computer Science and Engineering, Beihang University, China  
lizj@buaa.edu.cn

**Abstract.** In this paper, we propose a phrase reordering model for statistical machine translation. The model is derived from the bracketing ITG, and integrates the local and global reordering model. We present a method to extract phrase pairs from a word-aligned bilingual corpus in which the alignments satisfy the ITG constraint, and we also extract the reordering information for the phrase pairs, which are used to build the re-ordering model. Through experiments, we show that this model obtains significant improvements over the baseline on a Chinese-English translation.

**Keywords:** Statistical Machine Translation, Reordering model, ITG.

## 1 Introduction

Word and phrase reordering is very important in the Statistical Machine Translation (SMT) systems. In the state-of-the-art SMT system, it handles the word reordering through phrases, and then using the phrase reordering model and language model to restrict the order of the phrases. The origin phrase reordering model is the distortion model[1], which only considers the jump distances between the target phrases for adjacent source phrases. It is based on a simple assumption: the source language and the target language is monotone generally. For two languages which are close, such as French and English, this simple model may achieve good results; however, for two languages which are very different in word order, such as Chinese and English, the distortion model is not enough.

Many researchers have proposed different reordering models. Some of them[2][3] predicate the reorderings of the adjacent phrase pairs, and they only handle local reordering. And the others[4][5][6][7][8] handle the global reordering, i.e., they predicate the reordering of long distances. In the global models, Nagata's[4] clustered model predicates the reordering based on the current phrase pair and the previous phrase pair; Yamata[5] restricts the phrases order by the syntax tree, and Wu[6] proposes an Inversion Transduction Grammar (ITG), Chiang[7] presents a hierarchical model which uses a *formally* syntax-based model to constrain the order of the phrases. Xiong's[8] maximum entropy model is derived from the ITG model, which transforms the problem predicating the reordering into a classification problem.

In this paper, we propose a novel phrase reordering model, which is derived from bracketing ITG model and integrates the local and global reordering models. The rest of this paper is organized as follows: Section 2 presents how to derive the new reordering model from the bracketing ITG. In Section 3, we introduce how to train the models in our SMT model, consisting mainly of the translation models and the reordering model. In Section 4, we design the decoder. In Section 5, we test our model and compare it with the baseline system. Then, we conclude in Section 6.

## 2 Reordering Model

Since our reordering model is based on the bracketing ITG model, we will introduce the ITG model firstly, and then propose how to achieve our reordering model from it.

### 2.1 Bracketing ITG Model

Bracketing ITG is a synchronous context-free grammar, which generates two output streams simultaneously. It consists of the following five types of rules:

$$A \xrightarrow{a_{[]}} [AA] \quad (1)$$

$$A \xrightarrow{a_{<}} \langle AA \rangle \quad (2)$$

$$A \xrightarrow{b_{ij}} c_i / e_j \quad (3)$$

$$A \xrightarrow{b_{i\epsilon}} c_i / \epsilon \quad (4)$$

$$A \xrightarrow{b_{\epsilon j}} \epsilon / e_j \quad (5)$$

where  $A$  is the only non-terminal symbol,  $[]$  and  $\langle \rangle$  represent the two operations which generate outputs in straight and inverted orientation respectively.  $c_i$  and  $e_j$  are terminal symbols, which represent the words in both languages,  $\epsilon$  is the null words. The  $a_{[]}$ ,  $a_{<}$ ,  $b_{ij}$ ,  $b_{i\epsilon}$  and  $b_{\epsilon j}$  are the probabilities of the rules. The last three rules are called lexical rules.

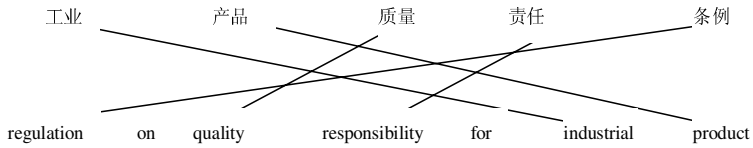
In this paper, we consider the phrase-based SMT, so the  $c_i$  and  $e_j$  represent phrases in both languages, which are consecutive words. The number of words within the phrase is called the length of the phrase. And a pair of  $c_i$  and  $e_j$  is called a phrase-pair, or a block. In rules (1) and (2), the  $A$  in the left side is composed of the two  $A$ s in the right side, so we call the left  $A$  parent block, and the right  $A$  child block. The block generated from lexical rules is called an atom block.

During the process of decoding, each phrase  $c_i$  in the source sentence is translated into a target phrase  $e_j$  through lexical rules, and then rules (1) or (2) are used to merge two adjacent blocks into a large block in straight or inverted orientation, until

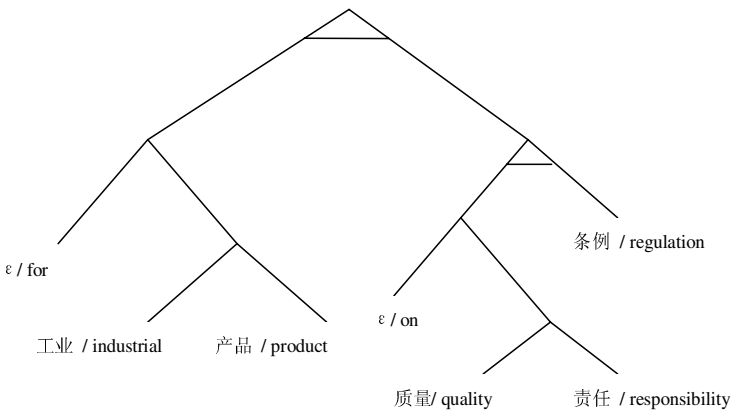


the whole source sentence is covered. In this way, we will obtain a binary branching tree, which is different from the traditional syntactical tree, and in which each constituent is a block, which only needs to satisfy the consecutive constraint.

Since the ITG model only needs to preserve the constituent structure, it achieves a great flexibility to interpret almost arbitrary reordering during the decoding, while keeping a weak but effective reordering constraint in the global scope. Figure 1 gives an example to illustrate a derivation from the ITG model.



(a) A word alignment



(b) An ITG tree

**Fig. 1.** (a) A word alignment and (b) an ITG tree which is derived from the ITG. A line between the branches means an inverted orientation, otherwise a straight one.

On the other hand, the  $a_{\square}$  and  $a_{\diamond}$  in the rules (1) and (2) are independent of the blocks in the right side, they only represent the preference to choose straight or inverted orientation. Thus, it is hard to predict the local reorderings of adjacent blocks.

In this paper, we hope to find an approach which will strengthen the model's ability to predict the local reorderings while keeping the global constraint.

### 2.2 New Reordering Model

Our problem is to predicate the reordering  $o \in \{straight, inverted\}$  of any two adjacent blocks  $A^1$  and  $A^2$ , we use  $O(A^1, A^2)$  to represent it, and use  $r(o, A^1, A^2)$  to

represent the probability that  $O(A^1, A^2) = o$ . A straight-forward method is to compute the co-occurrence count between  $A^1$  and  $A^2$ , and the frequencies they are in straight or inverted orientation respectively. And then use the MLE to predicate the reordering probabilities.

However, due to the constraints for corpus size and the memory of the computer, it is impossible to collect the reorderings of any two blocks; and in general, the larger the block is, the smaller the frequency it occurs, so that the reordering probabilities are not accurate.

So, instead of recording all the reorderings of any two blocks, we predicate the reorderings of each atom block  $A^o$  and any other block  $A^*$  which is preceding or posterior to the  $A^o$ . Generally, an atom block is shorter and the count it occurs will be larger, so the predication will be more accurate. For each atom block  $A^o$ , there are the following four reorderings:

1.  $a_{\square*}^o$  : the probability that  $O(A^o, A^*) = straight$ .
2.  $a_{\diamond*}^o$  : the probability that  $O(A^o, A^*) = inverted$ .
3.  $a_{*\square}^o$  : the probability that  $O(A^*, A^o) = straight$ .
4.  $a_{*\diamond}^o$  : the probability that  $O(A^*, A^o) = inverted$ .

Now, deriving from the rules (1) and (2), we will be able to predicate the reorderings of any two adjacent blocks  $A^1$  and  $A^2$ .

1. If  $A^1$  and  $A^2$  are both atom blocks, then

$$r(straight, A^1, A^2) = a_{\square*}^1 \bullet a_{*\square}^2 \tag{6}$$

$$r(inverted, A^1, A^2) = a_{\diamond*}^1 \bullet a_{*\diamond}^2 \tag{7}$$

I.e., if the reordering of  $A^1$  and  $A^2$  is straight, the probability is the product of the  $a_{\square*}^1$  and  $a_{*\square}^2$ , where  $a_{\square*}^1$  is the probability that  $O(A^1, A^*) = straight$  and  $a_{*\square}^2$  is the probability that  $O(A^*, A^2) = straight$ . In the same way, if the reordering of  $A^1$  and  $A^2$  is inverted, the probability is the product of the  $a_{\diamond*}^1$  and  $a_{*\diamond}^2$ .

2. If  $A^1$  or  $A^2$  are not atom blocks, then we can always find the rightest child block  $A_o^1$  of  $A^1$  which is an atom block, and the leftest child block  $A_o^2$  of  $A^2$  which is an atom block. And then we use formula (6) and (7) to predicate the  $r(o, A_o^1, A_o^2)$ . That is, we use the two adjacent atom blocks within  $A^1$  and  $A^2$  respectively to represent the whole blocks.

In this way, we can predicate the  $r(o, A^1, A^2)$  of any two adjacent blocks, which can be small or large, if only the blocks satisfy the constituent structure. During decoding, a sequence of application of rules (1) and (2) is the process of phrase reordering, so we define the reordering model as follows:

$$\Pr(O) = \prod_i r(o_i, A^{i1}, A^{i2}) \quad (8)$$

where  $r(o_i, A^{i1}, A^{i2})$  is the probability to apply the rules (1) or (2) in the  $i$ -th time.

### 3 Building the Model

The state-of-the-art SMT model is the log-linear model[9] :

$$\Pr(E|C) = \frac{\exp[\sum_{m=1}^M \lambda_m h_m(E, C)]}{\sum_{E'} \exp[\sum_{m=1}^M \lambda_m h_m(E', C)]} \quad (9)$$

where  $h_m(E, C)$  represents the features and  $\lambda_m$  is the weight of the feature  $h_m(E, C)$ .

In our model, we apply mainly the following features:

- Language model[10]  $P_{lm}(e)$ . We use a trigram language model of the target language to predicate the target words in block  $A^2$  while given the final two target words in block  $A^1$ .
- Translate models  $P(e|c)$ ,  $P(c|e)$ ,  $P_w(e|c)$ ,  $P_w(c|e)$ . The first two models are phrase translation models, and the last two are lexical translation models.
- Reordering model  $\Pr(O)$ . The model predicates the reordering of two adjacent blocks. For the reordering model is based on bracketing ITG, so the decoder must satisfy the ITG constraint. In this paper, the process of decoding is a sequence of applying the rules (1)-(5).

#### 3.1 Collection of Blocks

In order to train the translate models and the reordering model, we use a word-aligned bilingual corpus, in which the word alignments satisfy the ITG constraint. Figure 1 illustrates a valid word alignment example, which satisfies the ITG constraint, and forms a binary branching tree, in which the leaves represent the aligned word pair. In our word alignment, each word pair may consist of multi words in both sentences, but they must be consecutive.

For the word alignment can form a hieratical binary tree, we can extract the blocks in a straight-forward way, i.e. choosing each constituent as a block. Due to the memory constraint, we restrict the maximum length  $N$  of each block. In this paper, we set the  $N = 5$ . Because we need compute the lexical translation model and reordering model, we collect the following information at the same time when collecting each block  $A^o$  :

1. The word alignment within the block  $A^o$ . For the word alignment is hieratical, i.e. each constituent may be a leaf or consist of two child constituents. We record the information whether the constituent is a leaf or the division of the child constituents.
2. The reordering of the block  $A^o$  and the preceding or posterior blocks.

Thus, the final information of each block  $A^o$  consists of the block text, frequency of the block, lexical alignment, and the reordering. Table 1 lists the blocks which are extracted from the word alignment example in Figure 1. After collecting all the blocks in each word alignment in the trained bilingual corpus, we combine the same blocks and obtain the final block table.

**Table 1.** The blocks extracted from word alignment in Figure 1, the number in reordering column represents respectively the frequency of the reordering of the block  $A^o$  and the preceding and posterior blocks

Chinese Text	English Text	Frequency	Alignment	Reordering
工业	industrial	1	1-1	1 0 1 0
产品	product	1	1-1	1 0 0 1
工业产品	industrial product	1	1-1; 2-2	1 0 0 1
工业产品	for industrial product	1	1-2; 2-3	1 0 0 1
质量	quality	1	1-1	0 1 1 0
责任	responsibility	1	1-1	1 0 0 1
质量责任	quality Responsibility	1	1-1;2-2	0 1 0 1
质量责任	on quality responsibility	1	1-2;2-3	0 1 0 1
条例	regulation	1	1-1	0 1 1 0
质量责任 条例	regulation on quality responsibility	1	1-3;2-4;3-1	1 0 1 0

### 3.2 Translation Models

There are four translation models, we can obtain each model easily using the block table by MLE:

$$p(e|c) = \frac{\text{frequency of the block}(c, e)}{\sum_{e'} \text{frequency of the block}(c, e')}$$

$$p(c|e) = \frac{\text{frequency of the block}(c, e)}{\sum_{c'} \text{frequency of the block}(c', e)}$$

We can smooth the models using some smoothing algorithms, such as Simple Good-Turing Smoothing (SGT). For each block may be a leaf or consist of two child blocks,

$$p_w(e|c) = \begin{cases} p(e|c) & \text{if } (c, e) \text{ is a leaf} \\ p_w(e_1|c_1) \bullet p_w(e_2|c_2) & \text{otherwise} \end{cases}$$

$$p_w(c|e) = \begin{cases} p(c|e) & \text{if } (c, e) \text{ is a leaf} \\ p_w(c_1|e_1) \bullet p_w(c_2|e_2) & \text{otherwise} \end{cases}$$

I.e., we can obtain the lexical translation model hieratically.

### 3.3 Reordering Model

The block table also contains the reordering information for each block, which consists of the frequencies in straight and inverted orientation. So, we can obtain the reordering model by MLE in the same way:

$$a_{\square}^{ce} = \frac{\text{frequency in the straight orientaion with the posterior block}}{\text{frequency of the block } (c, e)}$$

$$a_{\diamond}^{ce} = \frac{\text{frequency in the inverted orientaion with the posterior block}}{\text{frequency of the block } (c, e)}$$

$$a_{\square}^{ce} = \frac{\text{frequency in the straight orientaion with the preceeding block}}{\text{frequency of the block } (c, e)}$$

$$a_{\diamond}^{ce} = \frac{\text{frequency in the inverted orientaion with the preceeding block}}{\text{frequency of the block } (c, e)}$$

## 4 Decoder

Our SMT model applies a reordering model which is based on the ITG model, so that, when given a source sentence  $C$ , the decoder must generate a target sentence  $E$ , which satisfies the ITG constraint, i.e.,  $C$  and  $E$  form a hieratical binary branching tree.

We developed a CKY style decoder with beam search, which searches the best  $E^*$  which holding the ITG constraint, when given a source sentence  $C$ . The following codes show the detail of the decoder.

$$E^* = \arg \max_E \{\Pr(E|C)\} = \arg \max_E \{\exp[\sum_{m=1}^M \lambda_m h_m(E, C)]\} \quad (10)$$

## The pseudo-code of the decoder

```

1: Function Decode()
2: Input:  sourceSent, srcLen
3: Output: targetSent
4: Begin
5:   for i=1 to srcLen
6:     for j=1 to srcLen
7:       foreach block A in block Table
8:         if A=(cij, e) then
9:           Add_Cand(cands[i][j], A, score(A))

10:  for j=2 to srcLen
11:    for i=1 to srcLen - j + 1
12:      for k=1 to j-1
13:        foreach cand A in cands[i,k]
14:          foreach cand B in cands[i+k,j-k]
15:            prob =score(A)*score(B) * po ([AB]) * plm ([AB])
16:            Add_Cand(cands[i][j], [AB], prob)
17:            prob =score(A)*score(B) * po (<AB>) * plm (<AB>)
18:            Add_Cand(cands[i][j], <AB>, prob)

19:  sort(cands[1][srcLen])
20:  return cands[1][srcLen][0]
21: End

```

During decoding, we keep a list for each cell  $(i,j)$ , and each element in the list is a candidate translation for the source phrase  $c_{ij}$ . The candidate consists of three type of information: the block text, the score of the candidate, and the reordering. In lines 15-18, we obtain the new larger block by combining the two child blocks in straight or inverted orientation.

We prune the search space in two ways. First, we keep the  $n$ -best candidates for each phrase and remove the others; second, we set a value  $\alpha$ , if the score of one candidate is worse than the  $\alpha$  times the best candidate score, the candidate is removed. Here, we set  $n=10$ ,  $\alpha=0.1$ .

Before testing, we apply the minimum error rate method to learn the weights of the features.

## 5 Experiments

We carried out experiments on a Chinese-English bilingual corpus, and compared with the state-of-the-art distortion-based decoder Moses<sup>1</sup>, which is an extension to the Pharaoh[11]. Table 2 shows the statistics of the training corpus, development corpus and test corpus.

For the baseline system, we used the default features: language model, translate models which were similar to ours, distortion model, word penalty and phrase

---

<sup>1</sup> <http://www.statmt.org/moses/>

**Table 2.** The statistic of the corpus

		Chinese	English
Training Corpus	Sentences	43,021	43,021
	Words	963,972	1,305,114
	Vocabulary	13,497	12,258
Develop Corpus	Sentences	300	300
	Words	4,774	6,250
Test Corpus	Sentences	300	300
	Words	7,613	9,760

penalty. We ran the default trainer in the Moses to train all of models and tune the feature weights, in which the trainer used the Giza++[12] to achieve word alignment and the minimum error rate approach to tune the weights. And then we ran the decoder in the Moses on the test set.

For our SMT model, after obtaining the word alignment for each sentence pair in the corpus, which satisfied the ITG constraint, we collected the atom blocks and built the block table. And then we trained the translation models and reordering model, and tuned the feature weights, and ran the decoder in section 4.

**Table 3.** Results on baseline system and our system, with and without reordering model

System	Bleu (%)
Moses	25.32
Ours - reordering model	25.65
Ours + reordering model	27.29
Ours + n-best word alignment	28.82

The results are listed in Table 3. The first column lists our MT systems with or without the reordering model; the second column lists the Bleu scores for the MT systems. The Bleu score computes the ratio of the n-gram for the translation results found in reference translations[13], and the higher scores indicates the better translations.

The second line is the result of the baseline system, the third line is the result of our system which does not apply the reordering model, the fourth line is the result of our system which applies the reordering model, and the last line shows the result of our system which obtain the block table from the n-best word alignments for each sentence pair.

From the Table 3, we observed that the results of the baseline system and ours without reordering model are close for they applied the similar features except that we obtained the block table in a different way. And our system with reordering model achieves a significant improvement (about 8%) over the baseline system.

In addition, the system with reordering model and n-best word alignments obtains the best result, which is about 5% improvement over the system with reordering

model. We conclude that it is because our training corpus is small that the block table obtained from n-best word alignment will cope with the sparseness partially.

## 6 Conclusion

The phrase reordering models in recent SMT may be divided into two categories, one is local model which predicates the local reordering of the adjacent blocks, and the other is global model, which can predicate the reordering of long distances. In this paper, we proposed an ITG-based reordering model, which can integrate the local and global model. Our model explicitly models the reordering of each atom block  $A^o$  and any blocks which are preceding or posterior to  $A^o$ , and then by satisfying the ITG constraint, we may predicate the reordering of any two blocks which may be small or large. Experiments on a Chinese-English bilingual corpus showed that our reordering model achieves an improvement over the baseline system.

In the future, we will test our model on the larger corpus and consider how to expand the reordering model to incorporate more information, for our reordering model only uses the two adjacent atom blocks, which are respectively the child blocks of the two larger blocks, to predict the reordering of the parent blocks.

## References

1. Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L.: The Mathematics of Statistical Machine Translation: Parameter estimation. *Computational Linguistics* 19(2), 263–312 (1993)
2. Tillmann, C., Zhang, T.: A Localized Prediction Model for Statistical Machine Translation. In: *Proceedings of the 43rd Annual Meeting of the ACL*, pp. 557–564 (2005)
3. Kumar, S., Byrne, W.: Local Phrase Reordering Models for Statistical Machine Translation. In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pp. 161–168 (2005)
4. Nagata, M., Saito, K.: A Clustered Global Phrase Reordering Model for Statistical Machine Translation. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pp. 713–720 (2006)
5. Yamada, K., Knight, K.: A Syntax-based Statistical Translation Model. In: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 523–530 (2001)
6. Wu, D.: Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics* 23(3), 374 (1997)
7. Chiang, D.: A Hierarchical Phrase-Based Model for Statistical Machine Translation. In: *Proc. of ACL 2005*, pp. 263–270 (2005)
8. Xiong, D., Liu, Q., Lin, S.: Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pp. 521–528 (2006)
9. Joseph Och, F., Ney, H.: Discriminative training and maximum entropy models for statistical machine translation. A Systematic Comparison of Various Statistical Alignment Models. In: *Proceedings of the 40th Annual Meeting of the ACL*, pp. 295–302 (2002)



10. Stolcke, A.: SRILM – An Extensible Language Modeling Toolkit. In: Proceedings of the International Conference on Spoken Language Processing, vol. 2, pp. 901–904 (2002)
11. Koehn, P.: Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In: Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas, pp. 115–124 (2004)
12. Joseph Och, F., Ney, H.: A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29(1), 19–52 (2003)
13. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311–318 (2002)

# Hobbs' Algorithm for Pronoun Resolution in Portuguese

Denis Neves de Arruda Santos and Ariadne Maria Brito Rizzoni Carvalho

Institute of Computing, State University of Campinas,  
Caixa Postal 6176, 13083-970, Campinas, SP, Brazil  
denis.santos@students.ic.unicamp.br, ariadne@ic.unicamp.br

**Abstract.** Automatic pronoun resolution may improve the performance of natural language systems, such as translators, generators and summarizers. Difficulties may arise when there is more than one potential candidate for a referent. There has been little research on pronoun resolution for Portuguese, if compared to other languages, such as English. This paper describes a variant of Hobbs' syntactic algorithm for pronoun resolution in Portuguese. The system was evaluated comparing the results with the ones obtained with another syntactic algorithm for pronoun resolution handling, the Lappin and Leass' algorithm. The same Portuguese corpora were used and significant improvement was verified with Hobbs' algorithm.

## 1 Introduction

Anaphora is a linguistic phenomenon of making an abbreviated reference to some entity (or entities) expecting the perceiver of the discourse to be able to disabbreviate the reference and, therefore, determine the identity of the entity. The abbreviated reference is called an anaphor, and the entity to which it refers is its referent or antecedent. The process of determining the referent of an anaphor is called resolution [8]. One of the most common types of anaphora is pronominal anaphora. Difficulties may arise when there is more than one potential candidate for a referent. Consider the following sentence:

- (1) *João culpou Pedro por ter batido seu carro.*  
[John blamed Peter for crashing his car.]

The anaphor “*seu*” (his) may either refer to “*João*” (John) or to “*Pedro*” (Peter).

Automatic pronoun resolution may improve the performance of natural language systems, such as translators, generators and summarizers. Several algorithms have been proposed to deal with the problem, such as Hobbs' algorithm, [9], Centering algorithm [7], and Lappin and Leass' algorithm [10]. Hobbs developed two approaches to pronoun resolution: a syntactic and a semantic approach.

In this paper we describe a variant of Hobbs' syntactic algorithm for pronoun resolution in Portuguese. Hobbs' algorithm was chosen due to its simplicity and good performance with English texts. The system was evaluated comparing the

results with the ones obtained with Lappin and Leass' algorithm. The same Portuguese corpora were used with both algorithms, and better performance was verified with Hobbs' algorithm.

The remainder of the paper is organized as follows: in Sect. 2, an overview on pronoun resolution is given; in Sect. 3, a variant of Hobbs' algorithm, followed working examples, is presented; in Sect. 4, Hobbs' algorithm is evaluated on three different corpora, and a comparison with Lappin and Leass' algorithm is made; finally, in Sect. 5, the conclusions and future work are presented.

## 2 Related Work

There have been a few approaches to pronoun resolution in Portuguese. In [14], the authors have proposed an algorithm for possessive pronoun handling in Portuguese. Their strategy was to use syntactic, semantic and pragmatic knowledge. The algorithm was evaluated using the text of Brazilian Laws on Environment Protection and, according to the authors, it succeeded in 92.97% of the cases.

In [1], the authors have evaluated the Centering algorithm for pronoun resolution in Portuguese texts. Centering algorithm [7] is based on a system of rules and restrictions that govern the relations between referring expressions. The authors used 16 juridical texts from the Attorney General's Office of the Republic of Portugal to evaluate the system and, according to them, the algorithm succeeded in 51% of the cases.

In [4,5] a variant of Lappin and Leass' algorithm for pronoun resolution in Portuguese was proposed. Lappin and Leass' algorithm [10] deals with third person pronouns in English; it uses a syntactic representation generated by a parser, and salience measures derived from the syntactic structure of the sentence. The algorithm was evaluated on three different corpora: 14 texts from a Brazilian magazine; a well-known literary book called "O Alienista" [2]; and the same 16 juridical texts used to evaluate the Centering algorithm [1]. For the magazine texts, the algorithm succeeded in 43.56% of the cases; for the literary book, the success rate was 32.61%; and, for the juridical texts, the algorithm succeeded in 35.15% of the cases.

In [13] the authors have proposed an algorithm for identifying noun phrase antecedents of third person personal pronouns, demonstrative pronouns, reflexive pronouns, and omitted pronouns in unrestricted Spanish texts. The authors defined a list of constraints and preferences for different types of pronominal expressions. The algorithm was evaluated on a corpus of 1,677 pronouns and achieved a success rate of 76.8%. The authors implemented other four competitive algorithms, including Hobbs' algorithm, and tested their performance on the same test corpus. The proposed algorithm obtained the best results.

## 3 Hobbs' Syntactic Algorithm

The syntactic algorithm developed by Hobbs solves intra- and inter-sentential pronominal anaphora. The strategy is a special traversal of the surface parse

tree<sup>1</sup>, looking for a noun phrase of the same gender and number of the pronoun [9]. The algorithm also deals with cataphora; however, it does not handle reflexive pronouns, and sentences which are themselves antecedents.

The tree is traversed in a breadth-first, left-to-right manner, because it is more likely that pronouns refer to subjects, instead of objects.

For convenience, Hobbs [9] makes the assumption that pronouns are immediately dominated by an NP node; hence, possessive nouns and pronouns have the structure illustrated in Fig. 1.

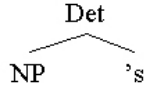


Fig. 1. Structure of a possessive noun or pronoun

Other assumption made by Hobbs is that an N within an NP node may have a prepositional phrase attached to it. This assumption is made to distinguish between the following two sentences:

- (2) Mr. Smith saw a driver in his truck.
- (3) Mr. Smith saw a driver of his truck.

In (2), pronoun “his” may either refer to the driver or to Mr. Smith; however, in (3) it can only refer to Mr. Smith. Figures 2(a) and 2(b) illustrate the corresponding parse trees for (2) and (3), respectively.

### 3.1 The Algorithm

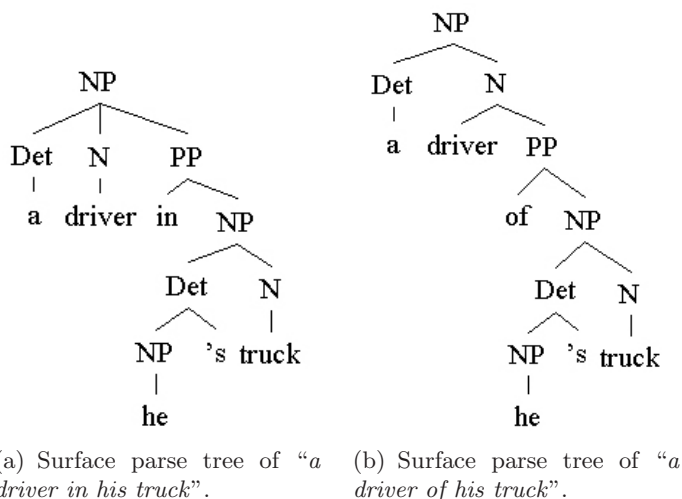
Hobbs’ algorithm is based on a particular traversal of the parse tree, looking for a noun phrase of the correct gender and number. Some constraints on anaphora resolution are encoded into the control structure of the algorithm [15]. The original algorithm was extended to deal with reflexive pronouns.

The algorithm takes as input a pronoun and a parse tree, and returns an antecedent to the pronoun. We will say that a node is *acceptable* if it agrees in gender and number with the pronoun. The variant of the algorithm is stated as follows, with the convention that all tree traversals are in left-to-right, breadth-first order:

1. Starting at the NP node which immediately dominates the pronoun, go up the tree to the first NP or S node encountered. Call this node X, and call the path used to reach it p.
2. (Reflexive case) If the pronoun is reflexive, traverse all branches below node X to the left of path p, and return the first acceptable NP node found.

<sup>1</sup> The tree that exhibits the grammatical structure of the sentence.

<sup>2</sup> Adv: adverb; Art: article; Conj: conjunction; Det: determiner; N: noun; NP: noun phrase; PP: prepositional phrase; Prp: preposition; S: sentence; VP: verb phrase.



**Fig. 2.** The corresponding parse trees for (2) and (3)

3. Traverse all branches below node  $X$  to the left of path  $p$  and return the first acceptable  $NP$  node encountered which has an  $NP$  or  $S$  node between it and  $X$ .
4. While  $X$  is not the highest  $S$  node in the sentence, repeat:
  - (a) From node  $X$ , go up the tree to the first  $NP$  or  $S$  node encountered. Call this new node  $X$ , and call the path traversed to reach it  $p$ .
  - (b) If  $X$  is an acceptable  $NP$  node, and if the path  $p$  to  $X$  did not pass through the  $N$  node that is immediately dominated by  $X$ , return  $X$ .
  - (c) Traverse all branches below node  $X$  to the left of path  $p$ , and return the first acceptable  $NP$  node found.
  - (d) If  $X$  is an  $S$  node, traverse all branches of node  $X$  to the right of path  $p$ , but do not go below any  $NP$  or  $S$  node encountered. Return the first acceptable  $NP$  node found.
5. Traverse the surface parse tree for the previous sentences in reverse order of occurrence proposing the first acceptable  $NP$  node found.

Reflexive pronouns are dealt with in step 2; step 5 is concerned with inter-sentential anaphora, that is, antecedents which belong to previous sentences. Hobbs [9] suggests to restrict the search to a five-sentence window. In step 4.b there is an embedded constraint that takes care of the case shown in Fig 2(b). Step 4.d handles the cases where the antecedent commands but does not precede the pronoun, that is, cataphora [9].

### 3.2 An Example

We follow the steps of the algorithm on three examples. Consider the following sentence, whose parse tree is shown in Fig 3.

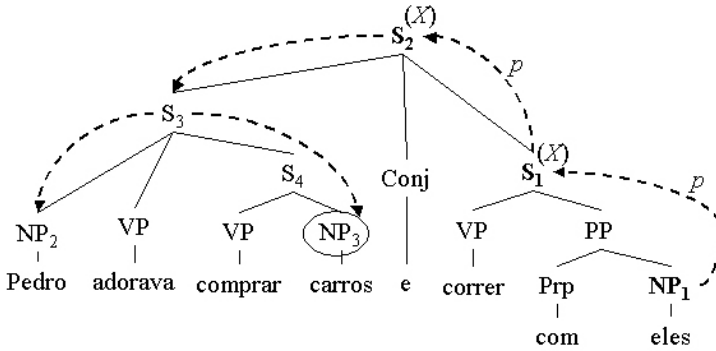


Fig. 3. Illustration of the algorithm

- (4) *Pedro adorava comprar carros e correr com eles.*  
 [Peter loved buying cars and racing them.]

According to step 1, execution begins at node NP<sub>1</sub>; node S<sub>1</sub> is visited and becomes node X. Since the pronoun is not reflexive, according to step 3, the left portion of S<sub>1</sub> tree is traversed, but no NP node is found. The loop in step 4 is entered, since S<sub>1</sub> is not the highest S node in the sentence. According to step 4.a, node S<sub>2</sub> is visited, and becomes node X. Step 4.b does not apply because node X (S<sub>2</sub>) is not an NP. According to step 4.c, the left portion of (S<sub>2</sub>) is traversed; and node NP<sub>2</sub> is the first NP found, but it is not acceptable (due to number). The next NP node in the traversal is NP<sub>3</sub>, which is returned.

Now consider (5), which contains a reflexive pronoun “se” (himself):

- (5) *O goleiro se machucou.*  
 [The goalkeeper hurt himself.]

The parse tree for (5) is shown in Fig. 4

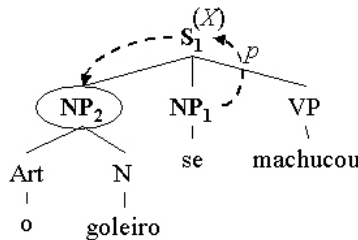


Fig. 4. Illustration of the algorithm with a reflexive pronoun

According to step 1, execution begins at node NP<sub>1</sub>; node S<sub>1</sub> is visited, and becomes node X. According to step 2, the left portion of S<sub>1</sub>'s tree is traversed, and node NP<sub>2</sub> is returned as antecedent.

Finally, consider (6), whose parse tree is shown in Fig 5. There are two possible candidates for referent, “João” and “Pedro”.

- (6) *João deu uma maçã para Pedro. Ele também deu uma melancia.*  
 [John gave an apple to Peter. He also gave an watermelon.]

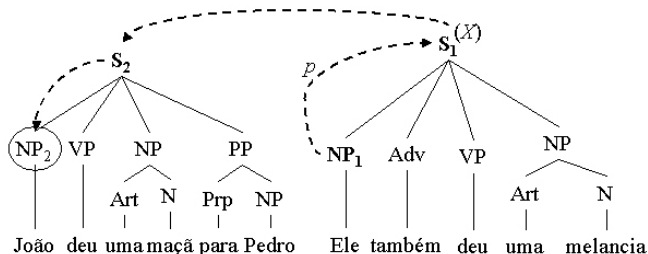


Fig. 5. Illustration of the algorithm with two possible referents

According to step 1, execution begins at node NP<sub>1</sub>; node S<sub>1</sub> is visited, and becomes node X. According to step 3, the left portion of the S<sub>1</sub> tree is traversed, but no node NP is found. Since S<sub>1</sub> is the highest node in the tree, the previous sentence is traversed, according to step 5, and node NP<sub>2</sub> is returned as antecedent.

## 4 Results

The algorithm was tested and evaluated on three different corpora. The first corpus was composed of 14 texts from a Brazilian magazine; the second was a literary book called *O Alienista*, from a well-known Brazilian author, Machado de Assis [2]; and the third was composed of 16 legal opinions from the Attorney General’s Office of the Republic of Portugal. The first corpus contained 196 pronouns, 92 being reflexive pronouns; the second corpus contained 634 pronouns, 119 being reflexive pronouns; and, finally, the last corpus contained 297 pronoun anaphora, with no reflexive pronouns.

The same corpora were used to evaluate the variant of Lappin and Leass’ algorithm and Hobbs’ algorithm. The parser PALAVRAS [3], a robust parser for Portuguese texts, was used to automatically annotate the corpora with morphosyntactic information. Pronouns were manually annotated with a tool for discourse annotation, MMAX (*Multi-Modal Annotation in Xml*) [12]. Another tool, Xtractor [6], was used to generate the XML encoding for the PALAVRAS output to improve the linguistic information extraction from the corpora analysed by PALAVRAS. The tool produces three XML files: *words*, which contains a list of the words from the text; *pos*, which contains morphosyntactic information; and *chunks*, which contains information on the text structure.

We have made some adjustment in the files generated by Xtractor and MMAX to make them suitable for the evaluation process.

In the experiments with the magazine and the literary corpus, we considered the resolution process successful if the solution offered by the algorithm was the same as that offered by manual annotation, or if there was a noun phrase which coreferred with the solution given by the annotators. Therefore, all the algorithm solutions were manually compared with the solution given by the annotators. With the juridical corpus, we considered the resolution process successful if the solution offered by our algorithm was the same as that offered by manual annotation. Therefore, all solutions given by the annotators were later automatically compared with the solutions given by our system.

Table 1 presents the results for the magazine texts; Table 2 presents the results for the literary book; and Table 3 presents the results for the juridical texts. The first column of the table shows the pronoun category (reflexive, non-reflexive and the total); the second shows the success rate for Hobbs' algorithm; and the third shows the success rate for Lappin and Leass' algorithm.

The results have shown that Hobbs' syntactic algorithm for pronoun resolution worked surprisingly well for Portuguese, despite its simplicity. This becomes more evident when compared with Lappin and Leass' algorithm, which is more elaborated. Our variant of Hobbs' algorithm performed significantly better than Lappin and Leass', except for a curious coincidence — the success rate of both algorithms for non-reflexive pronouns on the Brazilian magazine corpus was exactly the same.

Note the remarkable success achieved with the simple device of treating reflexive pronouns separately.

Compared with the results reported by Hobbs on English texts (88.3% of success rate), the numbers do not seem so impressive. Notice, however, that our

**Table 1.** Results on Brazilian magazine corpus

Pronoun	Quantity	Hobbs' algorithm	Lappin and Leass' algorithm
Reflexive	92	66 (71.74%)	35 (38.04%)
Non-reflexive	104	53 (50.96%)	53 (50.96%)
Total	196	119 (61.22%)	88 (44.90%)

**Table 2.** Results on literary corpus

Pronoun	Quantity	Hobbs' algorithm	Lappin and Leass' algorithm
Reflexive	119	82 (68.91%)	41 (34.45%)
Non-reflexive	515	233 (45.24%)	174 (33.79%)
Total	634	315 (49.68%)	215 (33.91%)

**Table 3.** Results on juridical corpus

Pronoun	Quantity	Hobbs' algorithm	Lappin and Leass' algorithm
Non-reflexive	297	120 (40.40%)	103 (35.15%)



tests were more extensive, on a substantially larger corpus. The complexity of available sentence forms in Portuguese should also be taken into account as a possible source of varying performance of the algorithm.

## 5 Conclusion and Future Work

We have developed a variant of Hobbs' algorithm for pronoun resolution in Portuguese. It was evaluated on three large corpora against Lappin and Leass' algorithm, and showed a markedly better performance.

As future work, we will introduce semantic knowledge through the application of selectional constraints. Hobbs reported an increase of accuracy to 91.7% with the use of semantics, and we believe a similar improvement will be obtained for Portuguese.

**Acknowledgments.** We thank CNPq for financial support. We are grateful to Thiago Thomes Coelho and Arnaldo Mandel for their valuable help.

## References

1. Aires, A.M., Coelho, J.C.B., Collovini, S., Quaresma, P., Vieira, R.: Avaliação de Centering em Resolução Pronominal da Língua Portuguesa. Taller de Herramientas y Recursos Lingüísticos para el Español y el Portugués. Iberamia (2004)
2. Assis, M.: O Alienista. VirtualBooks Literatura Brasileira. VirtualBooks (2002), [http://virtualbooks.terra.com.br/freebook/port/0\\_Alienista.htm](http://virtualbooks.terra.com.br/freebook/port/0_Alienista.htm)
3. Bick, E.: The parsing system PALAVRAS: Automatic grammatical analysis of Portuguese in a constraint grammar framework. Ártus University (2000)
4. Coelho, T.T.: Resolução de anáfora pronominal utilizando o algoritmo de Lappin e Leass. Master's thesis. University of Campinas (2005)
5. Coelho, T.T., Carvalho, A.M.B.R.: Lappin and Leass' Algorithm for Pronoun Resolution in Portuguese. In: Bento, C., Cardoso, A., Dias, G. (eds.) EPIA 2005. LNCS (LNAI), vol. 3808, pp. 680–692. Springer, Heidelberg (2005)
6. Gasperin, C.V., Vieira, R., Goulart, R.R.V., Quaresma, P.: Extracting XML chunks from Portuguese corpora. In: Proceedings of the Workshop on Traitement Automatique des Langues Minoritaires. Batz-sur-Mer (2003)
7. Grosz, B.J., Weinstein, S., Joshi, A.K.: Centering: A Framework for modeling the local coherence of discourse. In: Computational Linguistics, vol. 21, pp. 203–225. MIT Press, Cambridge (1995)
8. Hirst, G.: Anaphora in Natural Language Understanding. LNCS, vol. 119. Springer, Heidelberg (1981)
9. Hobbs, J.R.: Pronoun Resolution. Technical Report. City University of New York (1976)
10. Lappin, S., Leass, H.J.: An Algorithm for Pronominal Anaphora Resolution. In: Computational Linguistics, vol. 20, pp. 535–561. MIT Press, Cambridge (1994)
11. Mitkov, R.: Anaphora Resolution. Longman (2002)
12. Müller, C., Strube, M.: MMAX: A tool for the annotation of multi-modal corpora. In: The 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems IJCAI, pp. 45–50 (2001)

13. Palomar, M., Ferrández, A., Moreno, L., Martínez-Barco, P., Peral, J., Saiz-Noeda, M., Muñoz, R.: An algorithm for anaphora resolution in spanish texts. In: Computational Linguistics, vol. 27, pp. 545–567. MIT Press, Cambridge (2001)
14. Paraboni, I., Lima, V.L.S.: Possessive pronominal anaphor resolution in Portuguese written texts. In: The 17th International Conference on Computational Linguistics. Computational Linguistics, vol. 2, pp. 1010–1014. MIT Press, Cambridge (1998)
15. Rich, E., LuperFoy, S.: An architecture for anaphora resolution. In: Proceedings of the second conference on applied natural language. Computational Linguistics, pp. 18–24. MIT Press, Cambridge (1988)

# Automatic Acquisition of Attribute Host by Selectional Constraint Resolution

Jinglei Zhao, Hui Liu, and Ruzhan Lu

Department of Computer Science  
Shanghai Jiao Tong University  
800 Dongchuan Road Shanghai, China  
{zhaojl, lh.charles, rzlu}@cs.sjtu.edu.cn

**Abstract.** It is well known that lexical knowledge sources such as WordNet, HowNet are very important to natural language processing applications. In those lexical resources, attributes play very important roles for defining and distinguishing different concepts. In this paper, we propose a novel method to automatically discover the attribute hosts of HowNet's attribute set. Given an attribute, we model the solving of its host as a problem of selectional constraint resolution. The World Wide Web is exploited as a large corpus to acquire the training data for such a model. From the training data, the attribute hosts are discovered by using a statistical measure and a semantic hierarchy. We evaluate our algorithm by comparing the result with the original hand-coded attribute specification in HowNet. Some experimental results about the performance of the method are provided.

## 1 Introduction

Semantic lexicons such as WordNet[1], HowNet[2] etc. are key resources of knowledge for natural language processing tasks. However, manual construction of such knowledge bases is extremely labor-intensive and suffers from inconsistency and limited coverage. Consequently, there is a need for automatic methods of constructing or extending lexical knowledge bases.

In this paper, we explore to automatically construct and extend the attribute lexicon in HowNet[2]. HowNet is a hand-coded common-sense lexical knowledge base describing inter-conceptual relations and inter-attribute relations of concepts as connoting in lexicons of the Chinese and their English equivalents. The top-most level of classification in HowNet includes *entity*, *event*, *attribute* and *attribute value*. Under the *attribute* level, the lexicon contains 2093<sup>1</sup> different attribute words each with a specification as following:

(1) 寿命(life-span): attribute|属性, age|年龄, &animate|生物

in which, the first roll of the specification (attribute|属性) indicates that 寿命(*life-span*) is an attribute. The second roll gives that 寿命(*life-span*) belongs to the subcategory "age|年龄" of attribute. The third roll gives the concept class that can have the attribute 寿命(*life-span*). Such a concept class is named as the *host* of attribute in HowNet.

<sup>1</sup> In the version: HowNet 2002.

Given such a form of lexical specification in HowNet, the task of constructing and extending HowNet's attribute knowledge base can be naturally divided into three sub-tasks. First, acquiring a large set of attribute words. Second, structuring the attribute set into a hierarchy. Third, discovering the hosts of the extended attributes. In this paper, however, we focus on methods for the third subtask, that is, automatic discovering the attribute hosts of the attribute set.

We propose a novel algorithm for such an attribute host discovery task. Given an attribute, we model the solving of its host as a resolution problem of selectional constraint. Using a lexico-syntactic pattern (LSP), we first extract from the web a training set of  $\langle$  concept word, attribute word  $\rangle$  pairs. Then using the original semantic hierarchy of *entity* in HowNet, we create the space of candidate constraint classes of the attribute word. Finally, the host of the attribute is acquired by selecting the most appropriate concept class from the candidate space through a statistical measure called selectional association. To evaluate our method, we compare the automatic discovered attribute hosts with HowNet's original specification, the experimental results show that our method is very effective.

The remainder of the paper is organized as follows: Section 2 describes the related works. Section 3 gives definitional information of attribute and attribute host. Section 4 presents the detailed description of the attribute host resolution algorithm. Section 5 provides the experimental results. Finally, in Section 6, we give the conclusions and discuss future work.

## 2 Related Works

### 2.1 Lexical Knowledge Acquisition

Two main sources have been exploited for the task of lexical resource acquisition in general. One is MRD, the other is large scale corpus.

The definitional information in an MRD describes basic conceptual relations of words and is considered easier to process than general free texts. So, MRDs have been used as a main resource for deriving lexical knowledge from the beginning. [3, 4] constructed hyponymy hierarchies from MRDs. [5–7] extracted more semantic information from MRDs beyond hyponymy such as meronymy etc. The main difficulty of using MRDs as a knowledge source, as noted by [8], is that much definitional information of a word is inconsistent or missed. Corpus is another important source for lexical knowledge acquisition. [9] used lexical-syntactic patterns like "NP such as List" to extract a hyponymy hierarchy and [10, 11] acquired *part-of* relations from the corpus.

Data sparseness is the most notorious problem for acquiring lexical knowledge from the corpus. However, the World Wide Web can be seen as a large corpus[12–15]. [16, 17] proposed Web-based bootstrapping methods which mainly employed the interaction of extracted instances and lexical patterns for knowledge acquisition. However, in our algorithm for attribute host acquisition, the exploitation of the Web is novel in that we use it as a source to acquire the training data for selectional constraint resolution.

## 2.2 Attribute Knowledge Acquisition

Specific to attribute knowledge discovery, previous works have mainly concentrated on acquiring attributes specific to a given concept or instantiated object.

The methods used for identifying specific attribute information relative to a concept fall into two categories. One is to use mainly the layout information of web pages [18, 19], such as HTML tables and structural tags like TD, CAPTION etc., which can be clues of attributes for describing specific objects. However, such kind of methods suffers from the subjectivity of the web page writer. The other kind of method exploits lexicosyntactic patterns in which an object and its attributes often co-occur. [20] used the pattern "*the \* of the C [is|was]*" to identify attributes from the Web in English.

Overall, in attribute knowledge acquisition, most previous works have concentrated on the acquisition of the attributes given a concept or an instantiated object. No work has ever done reversely for the automatic discovery of attribute host specific to a given attribute, as we do it in this paper.

## 3 Attributes and Hosts

Concerning the notion of attribute, there have been infinitely long philosophical debates since even Aristotle. Also, no well-formed definition was given for the attribute host in HowNet's manual specification. To make our task more explicated, in this section, we give the definition of attributes and its corresponding hosts and justify our effort for attribute-centered host discovery.

First, we define the notion of attribute using Woods' linguistic test [21]:  $A$  is an attribute of  $C$  if we can say " $V$  is a/the  $A$  of  $C$ ", in which,  $C$  is a concept,  $A$  is an attribute of  $C$  and  $V$  is the value of the attribute  $A$ . For example, in "*brown is a color of dogs*", *color* is an attribute of concept *dog* with the value *brown*. For such a definition, several essentials need to be explicated. First, attributes are nouns. Second, attributes are defined on concepts and must allow the filling of values. If a  $C$  or  $V$  cannot be found to fit woods' test,  $A$  cannot be an attribute. Third, different concepts can have the same attribute. For example, besides *dog*, concepts such as *elephant*, *furniture* can also have the attribute *color*.

Next, given an attribute  $A$ , we define the attribute host  $H$  of  $A$  with respect to a concept type hierarchy  $T = \langle Type, \sqsubseteq \rangle$ , in which  $\sqsubseteq$  is a partial order on the set of concepts  $Type$  indicating hyponym relations. In the concept set  $Type$ , assume that  $Type_A \subseteq Type$  is the subset of concepts which can have the attributes  $A$ . Then the attribute host  $H$  of  $A$  with respect to  $T$  is defined as  $\sqcap Type_A$ , which is the least upper bound of the set  $Type_A$ . This notion of host can be shown by the attribute *color*. Because nearly every object in the world can have the attribute *color*, its attribute host will be a top level concept class in the hierarchy  $T$ . Actually, in HowNet's attribute lexicon, such a host is specified as *physical*, the super concept of *animate* and *inanimate*.

A large attribute knowledge base with the attribute host explicated will be of great use for both knowledge engineering and natural language processing. In knowledge engineering, because attribute information play very important roles for defining and differentiating various concepts and different concepts can have the same attributes, an attribute centered view will help very much for the engineers to select the most ap-

appropriate attributes for defining related concepts. Also, many natural language processing applications such as noun clustering, product opinion mining could benefit from attribute knowledge. [20] illustrated that enriching vector-based models of concepts with attribute information led to drastic improvements in noun clustering. [22] explored product attributes for mining product opinions.

However, attribute words are very productive in the language. New attribute names are forming day by day. For example, 点击率 (*click-rate*) is a recently formed attribute brought by the growth of internet. The attribute knowledge source available in Chinese as the attribute lexicon in HowNet includes only a small part of attributes possibly occurred in the text. So, to satisfy the need for the above applications, methods must be provided for automatically harvesting attributes and its corresponding attribute hosts.

## 4 Attribute Host Resolution

In this section, we give our algorithm for the automatic discovery of attribute host. Given a known attribute, we model the solving of its attribute host as a selectional constraint resolution problem. Selectional constraints are limitations on the applicability of natural predicates to arguments [23]. Most previous works on selectional constraint resolution are conducted for the verb predicates. For example, the verbal concept *eat* have the AGENT constraints for *human* and *animal*, and a PATIENT constraint for *food*. Similarly, given the definition of attribute host in the above section, a specific attribute can also be viewed as having constraint selection for attribute hosts. For example, while the attribute *financial-power* is strongly related to *organization* involved in marketing activities, the attribute *vital-capacity* is restricted to *human* and *animal*.

```

Given: An Attribute Set Attr.
      A training set of <concept word, attribute word>
        pairs extracted from the Web.
      An IS-A concept hierarchy.
Begin:
1. Create the space of candidate preference classes
   for each attribute.
2. Evaluate the appropriateness of the candidates
   by a statistical measure.
3. Select the most appropriate subset in the candidate
   space to express the selectional preference.
End.
Output: Attribute Hosts for every attribute in Attr.

```

**Fig. 1.** The Algorithm for Attribute Host Resolution

The algorithm is illustrated in Figure 1. First, given an attribute set, the algorithm needs an IS-A concept hierarchy and a training set of <concept word, attribute word> pairs for discovering the attribute hosts. In our experiment, we use HowNet's *entity* hierarchy. A part of top-level of which is illustrated in Figure 2.

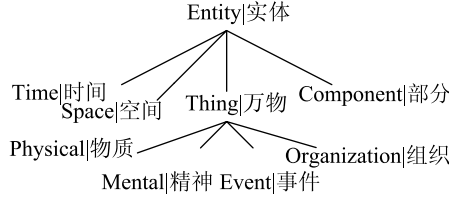


Fig. 2. Concepts in the Top Level Hierarchy of HowNet's Entity Category

Unlike most approaches of selectional constraint acquisition for verb predicates which mainly exploited parsed corpus for training set acquisition [24–26], we use a lexical pattern (a) to get the training set for attribute host discovery.

(a) N 的('s) A 很(very).

In the pattern (a), the Chinese character “很” (*very*) is a modifier indicating that words after them are values (such as adjectives) for the attribute. And the noun before the genitive “的” ('s) is expected to be a defining concept for the attribute. From the above aspects, we can see such a pattern is actually a variant of Wood's linguistic test for attribute definition and thus is well-motivated for acquiring the training set. Using such a pattern, given an attribute  $A$ , we instantiate the pattern and form a query to the search engine Google<sup>2</sup> and extract from the returning results of Google for the possible concept words (the noun  $C$  in the pattern) associated with  $A$ . For example, if the attribute is 创造力(*creativity*), we can extract from the Web using (a) for the possible concept words such as 青年(*young*), 员工(*staff*), 孩子(*children*), etc.

From a large set of such ⟨concept word, attribute word⟩ pairs, we generalize for the concept classes using HowNet's IS-A *entity* taxonomy and get the space of candidate constraint classes for each attribute. The appropriateness of the candidates is evaluated using the selectional association measure [23].

Formally, let  $A$  be a random variable ranging over the set  $a_1, \dots, a_m$  of attributes under consideration. Let  $C$  be a random variable ranging over the set  $c_1, \dots, c_n$  of classes in the taxonomy. The selectional association between an attribute  $a_i$  and a concept class  $c$  can be defined as:

$$Assoc(a_i, c) = \frac{P(c|a_i) \log \frac{P(c|a_i)}{P(c)}}{S(a_i)} \quad (1)$$

In which,  $S(a_i)$  is called the selectional preference strength of the attribute  $a_i$  that is modeled as the relative entropy between the probability distribution  $P(c|a_i)$  and  $P(c)$  as:

$$S(a_i) = \sum_c P(c|a_i) \log \frac{P(c|a_i)}{P(c)} \quad (2)$$

<sup>2</sup> www.google.com

The joint probabilities in the model were estimated from the training set of <concept word, attribute word> pairs as follows:

$$P(a_i, c) = \frac{1}{N} \sum_{n \in words(c)} \frac{1}{|classes(n)|} freq(a_i, n) \quad (3)$$

where  $words(c)$  denotes the set of nouns for which any sense is subsumed by  $c$ ,  $classes(n)$  denotes the set of classes to which noun  $n$  belongs.  $freq(a_i, n)$  is the number of times  $n$  appeared as the concept word of attribute  $a_i$  and  $N$  is the total number of instances in the extracted data set. After the evaluation of the association score of each candidate class, the most appropriate concept class in the candidate space is selected as the host for the attribute.

## 5 Experiments and Results

To test our model for the discovery of attribute hosts, we select 195 attributes from HowNet's attribute lexicon as our experimental data set. For such an attribute set, to train our model for attribute host resolution, we use the lexical pattern (a) in Section 4 to acquire the <concept word, attribute word> training pairs from the Web. Specifically, given an attribute, we first instantiate the pattern by the attribute to form a query to the Google search engine. For example, if the attribute is 速度(*speed*), the query formed would be " \*的速度很" ("\* 's speed is very "). Then we collect the top 1000 snippets from the returned searching result and segment and pos-tag the sentence where the query appear. From such pos-tagged sentence, we extract the set of concept words associated with the attribute under investigation. For the 195 attributes overall, we totally acquired 22,000 <concept word, attribute word> training pairs from the Google search engine through the previous steps. One point to note is that, when using such a training set and the concept class taxonomy illustrated in figure 2 to create the space of candidate classes, a threshold (set to two in our experiment) is used to ignore the possible noise introduced by the training set. That is, only those classes that have a higher number of the occurrences than the threshold are considered.

The selectional preference strength  $S(a_i)$  of an attribute  $a_i$  discussed in Section 4 can be an indicator for the attribute host's generality in the concept taxonomy. Table 1 lists a subset of attributes ranked in a descending order respect to their selectional preference strength. The behavior in Table 1 illustrates that the higher the strength, the more specific the attribute selects for its defining concepts. For example, the attribute 射程(*range-of-fire*) has a very high preference strength which means that its attribute host is very specific (restricted to weapon). Meanwhile, the attribute 状况(*state*) has a very low strength which means that its attribute host is very general (generalized to *entity*).

After computing the selectional preference strength for each attribute, the selectional association between an attribute and a specific candidate concept class is computed according to (1) in Section 4. Then from all the possible candidate classes of an attribute, the attribute host is discovered by selecting the concept class with which the attribute have the highest selectional association. Table 2 illustrates some examples of the hosts acquired for some attributes by our model. In which, for example, the host of



**Table 1.** Selectional Preference Strength of some Attributes

Attribute	Strength
烈度(earthquake-intensity)	5.24
库容(storage-capacity)	5.18
车速(vehicle-speed)	4.53
疗程(course-of-treatment)	4.24
冰点(freezing-point)	4.18
射程(range-of-fire)	3.93
口感(mouth-feel)	3.71
情况(situation)	0.13
规律性(regularity)	0.13
状况(state)	0.11
重要性(importance)	0.09
特殊性(particularity)	0.08
效应(effect)	0.03

attribute 气质(*temperament*) is 人(*human*) and the host of attribute 款式(*design*) is 人工物(*artifact*).

**Table 2.** Examples of HowNet's Specification

Attribute	Concept Words	Acquired Host
气质( <i>temperament</i> )	女孩( <i>girl</i> ), 男性( <i>man</i> )	人( <i>human</i> )
题材( <i>subject-matter</i> )	故事( <i>story</i> ), 艺术( <i>art</i> )	信息( <i>information</i> )
款式( <i>design</i> )	沙发( <i>settee</i> ), 商品( <i>commodity</i> )	人工物( <i>artifact</i> )
分子量( <i>molecular-weight</i> )	溶剂( <i>menstruum</i> ), 乙烯( <i>ethylene</i> )	物质( <i>physical</i> )
开本( <i>book-size</i> )	期刊( <i>periodical</i> ), 教材( <i>teaching-material</i> )	读物( <i>readings</i> )
态度( <i>attitude</i> )	法官( <i>judge</i> ), 服务员( <i>attendant</i> )	人( <i>human</i> )
音色( <i>tone-color</i> )	吉他( <i>guitar</i> ), 单簧管( <i>clarinet</i> )	乐器( <i>MusicTool</i> )
功用( <i>function</i> )	词典( <i>dictionary</i> ), 引擎( <i>engine</i> )	人工物( <i>artifact</i> )
文才( <i>literary-talent</i> )	作者( <i>author</i> ), 校长( <i>headmaster</i> )	人( <i>human</i> )
时效( <i>time-prescription</i> )	诉讼( <i>lawsuit</i> ), 纠纷( <i>dispute</i> )	事情( <i>affair</i> )

The quality of the acquired defining concepts of our experimental attribute set is compared to HowNet's hand-coded specification. Table 3 shows HowNet's original specification of the attributes listed in Table 2. Comparing the result of all the experimental attribute set with HowNet, Table 4 gives the number and the percentage of hosts which are exactly matched, not matched at all, or matched by more general or more specific classes in the taxonomy.

The results of the comparison is encouraging. About 39.0% of the resulted attribute hosts exactly match with HowNet's hand-coded definition and about 25.1% can match by 1 or 2 level of hyperonym or hyponym expansion. For example, our resulted defining concepts for 态度(*attitude*) is 人(*human*) while HowNet 动物(*animal-human*). Such an example is included in the case of "1 level of hyperonym" though we think that *human* is more appropriate than *animal*. Another example is the attribute 功用(*function*). Our

**Table 3.** Examples of Acquired Selectional Constraints of Attributes

Attribute	HowNet
气质(temperament)	人(human)
题材(subject-matter)	信息(information)
款式(design)	人工物(artifact)
分子量(molecular-weight)	物质(physical)
开本(book-size)	书刊(publications)
态度(attitude)	动物(AnimalHuman)
音色(tone-color)	声(sound)
功用(function)	实体(entity)
文才(literary-talent)	编辑(compile)
时效(time-prescription)	行动(act)

acquired host is 人工物(artifact), while 实体(entity) in HowNet. This case is included in ">2 level hyperonym". However, our automatically acquired host is reasonable given lexical theories such as generative lexicon [27], in which it is argued that the typical concept having the *telic(function)* attribute is *artifact*. A last example for explaining the experimental result is 音色(tone-color). For which, our acquired host is *music-tool*, while 声(sound) in HowNet. Such an example is included in "not matched".

**Table 4.** Comparison of Acquired Hosts with HowNet's Specification

exactly matched	76 (39.0%)
matched by 1 level hyperonym	21 (10.8%)
matched by 1 level hyponym	14 (7.1%)
matched by 2 level hyperonym	9 (4.6%)
matched by 2 level hyponym	5 (2.6%)
matched by >2 level hyperonym	24 (12.3%)
matched by >2 level hyponym	7 (3.6%)
not matched	37 (19.0%)

## 6 Conclusions and Future Work

In this paper, we have presented a novel method for automatically discovering the attribute host of a given attribute in natural language. We gave the definition of attribute host and modeled its acquisition as a problem of selectional constraint resolution. After that, we used a Woods' style lexical pattern and exploited the World Wide Web for acquiring a large training set of ⟨ concept word, attribute word ⟩ pairs. From such a training set, we applied an IS-A taxonomy to generalize the concept classes and used the selectional association measure to select out the attribute host from candidate classes. To evaluate our algorithm for attribute host acquisition, we compared the experimental result with the original hand-coded specification in HowNet. The comparison showed that our method was very effective.

However, much work need to be done in the future. Currently, we don't differentiate senses of the attribute words. About 8.5% of the attributes in HowNet's attribute lexicon have more than one nominal senses. In the future, we will study method to combine sense disambiguation with our current work to compute more precise hosts for attributes.

Also, although our current definition of attribute host is coincide with HowNet's intuitional specification where only one concept class is specified as the host for an attribute, it tends to generalize too much in some cases. For example, in both our result and the HowNet's specification, the host for the attribute 处境(*plight*) is 实体(*entity*). However, it may be more appropriate to represent it by a set of more specific concept classes, e.g. a set including 人(*human*), 场所(*InstituePlace*) and 团体(*Community*). In the future, we will probe such kind of amendments to the definition of attribute host and its corresponding acquisition algorithm.

Finally, in our algorithm, we use the hand-coded *entity* taxonomy in HowNet for attribute host acquisition. This approach can be viewed as making a closed world assumption for the concept class knowledge. However, as pointed out by some recent works in lexical knowledge acquisition [28], heterogenous relationships in lexical knowledge sources can influence each other. In the future, we will explore unsupervised methods and study the effects of dynamic changes of taxonomy information on the acquisition of attribute hosts.

## Acknowledgements

This work is supported by NSFC Major Research Program 60496326 :Basic Theory and Core Techniques of Non Canonical Knowledge.

## References

1. Fellbaum, C.: Wordnet: an electronic lexical database. MIT Press, Cambridge (1998)
2. Dong, Z., Dong, Q.: HowNet and the Computation of Meaning. World Scientific, Singapore (2006)
3. Amsler, R.A.: The Structure of the Merriam-Webster Pocket Dictionary (1980)
4. Chodorow, M.S., Byrd, R.J., Heidorn, G.E.: Extracting semantic hierarchies from a large on-line dictionary. In: Proceedings of the 23rd conference on Association for Computational Linguistics, pp. 299–304 (1985)
5. Wilks, Y., Fass, D., Guo, C., McDonald, J., Plate, T., Slator, B.: A tractable machine dictionary as a resource for computational semantics. Longman Publishing Group White Plains, NY, USA (1989)
6. Alshawi, H.: Analysing the dictionary definitions. Computational lexicography for natural language processing table of contents, 153–169 (1989)
7. Richardson, S.D., Dolan, W.B., Vanderwende, L.: MindNet: acquiring and structuring semantic information from text. In: Proceedings of the 17th international conference on Computational linguistics, pp. 1098–1102 (1998)
8. Ide, N., Veronis, J.: Extracting knowledge bases from machine-readable dictionaries: Have we wasted our time. Proceedings of KB&KS 93, 257–266 (1993)

9. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th conference on Computational linguistics, vol. 2, pp. 539–545 (1992)
10. Berland, M., Charniak, E.: Finding parts in very large corpora. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pp. 57–64 (1999)
11. Poesio, M., Ishikawa, T., im Walde, S.S., Viera, R.: Acquiring lexical knowledge for anaphora resolution. In: Proceedings of the 3rd Conference on Language Resources and Evaluation (LREC) (2002)
12. Grefenstette, G., Nioche, J.: Estimation of English and non-English Language Use on the WWW. Arxiv preprint cs.CL/0006032 (2000)
13. Jones, R., Ghani, R.: Automatically building a corpus for a minority language from the web. In: Proceedings of the Student Research Workshop at the 38th Annual Meeting of the Association for Computational Linguistics, pp. 29–36 (2000)
14. Zhu, X., Rosenfeld, R.: Improving trigram language modeling with the World Wide Web. In: Acoustics, Speech, and Signal Processing, 2001. Proceedings (ICASSP 2001). 2001 IEEE International Conference on, vol. 1 (2001)
15. Keller, F., Lapata, M.: Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics* 29(3), 459–484 (2003)
16. Brin, S.: Extracting patterns and relations from the world wide web. In: WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT 1998, pp. 172–183 (1998)
17. Pennacchiotti, M., Pantel, P.: A Bootstrapping Algorithm for Automatically Harvesting Semantic Relations. In: Proceedings of Inference in Computational Semantics (ICoS-2006), Buxton, England (2006)
18. Chen, H.H., Tsai, S.C., Tsai, J.H.: Mining tables from large scale html texts. In: 18th International Conference on Computational Linguistics (COLING), pp. 166–172 (2000)
19. Yoshida, M., Torisawa, K., Tsujii, J.: A method to integrate tables of the world wide web. In: Proceedings of the International Workshop on Web Document Analysis (WDA 2001), Seattle, US (2001)
20. Almuhareb, A., Poesio, M.: Attribute-Based and Value-Based Clustering: An Evaluation. In: Proc. of EMNLP, pp. 158–165 (2004)
21. Woods, W.: What's in a Link: Foundations for Semantic Networks. Bolt, Beranek and Newman (1975)
22. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of EMNLP 2005 (2005)
23. Resnik, P.: Selectional constraints: an information-theoretic model and its computational realization. *Cognition* 61(1-2), 127–159 (1996)
24. Framis, F.R.: An experiment on learning appropriate Selectional Restrictions from a parsed corpus. In: Proceedings of the 15th conference on Computational linguistics, vol. 2, pp. 769–774 (1994)
25. Ribas, F.: On learning more appropriate selectional restrictions. In: Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics, pp. 112–118 (1995)
26. Wagner, A.: Enriching a lexical semantic net with selectional preferences by means of statistical corpus analysis. In: Proceedings of the ECAI-2000 Workshop on Ontology Learning, pp. 37–42 (2000)
27. Pustejovsky, J.: *The Generative Lexicon*. Bradford Books (1998)
28. Snow, R., Jurafsky, D., Ng, A.: Semantic taxonomy induction from heterogenous evidence. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL, pp. 801–808 (2006)

# E-Gen: Automatic Job Offer Processing System for Human Resources

Rémy Kessler<sup>1,3</sup>, Juan Manuel Torres-Moreno<sup>1,2,\*</sup>, and Marc El-Bèze<sup>1</sup>

<sup>1</sup> Laboratoire Informatique d'Avignon, BP 1228 F-84911 Avignon Cedex 9 France  
{remy.kessler,juan-manuel.torres,marc.elbeze}@univ-avignon.fr  
<http://www.lia.univ-avignon.fr>

<sup>2</sup> École Polytechnique de Montréal - Département de génie informatique  
CP 6079 Succ. Centre Ville H3C 3A7, Montréal (Québec), Canada

<sup>3</sup> AKTOR Interactive Parc Technologique 12, allée Irène Joliot Curie Bâtiment B3  
69800 Saint Priest France

**Abstract.** The exponential growth of the Internet has allowed the development of a market of on-line job search sites. This paper aims at presenting the E-Gen system (Automatic Job Offer Processing system for Human Resources). E-Gen will implement two complex tasks: an analysis and categorisation of job postings, which are unstructured text documents (e-mails of job listings possibly with an attached document), an analysis and a relevance ranking of the candidate answers (cover letter and *curriculum vitae*). This paper aims to present a strategy to resolve the first task: after a process of filtering and lemmatisation, we use vectorial representation before generating a classification with Support Vector Machines. This first classification is afterwards transmitted to a "corrective" post-process which improves the quality of the solution.

## 1 Introduction

The exponential growth of the Internet has allowed the development of an on-line job-search sites market [1,2,3]. The mass of information obtained through candidate response represents a lot of information that is difficult for companies to manage [4,5,6]. It is therefore indispensable to process this information by an automatic or assisted way. The *Laboratoire Informatique d'Avignon* (LIA) and Aktor Interactive have developed the E-Gen system in order to resolve this problem. It will be composed of two main modules:

1. A module to extract information from a corpora of e-mails containing job descriptions.
2. A module to analyse and compute a relevance ranking of the candidate answers (cover letter and *curriculum vitae*).

In order to extract useful information, the system analyses the contents of the e-mails containing job descriptions. In this step, there are many difficulties and

---

\* Corresponding author.

interesting problems to resolve related to Natural Language Processing (NLP), for example, that job postings are written in free-format, strongly unstructured, with some ambiguities, typographic errors, etc. Similar work has been carried out in the recruitment domain [17], but this concerns only the handling of responses and not integration of job offers.

Aktor Interactive<sup>1</sup> is a French communication agency, specialised in e-recruiting. Aktor's key service is the publication of job adverts on different online job boards on behalf of their clients. Therefore a system that is able to automate this is desirable due to the high number and spread of specialised<sup>2</sup>, non-specialised<sup>3</sup> or still local sites<sup>4</sup>. To do this, Aktor uses an automatic system to send job offers in XML format (Robopost Gateway) defined with job boards. Therefore, the first step of the workflow is to identify every part of the job posting and extract the relevant information (contract, salary, localization etc.) from the received job offer. Figure 1 shows an overview of the workflow. Until now, the first step of the workflow was a laborious manual task: with users having to copy and paste job offers in the Aktor Information system.

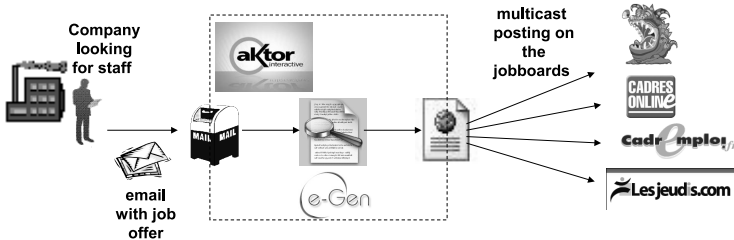


Fig. 1. Aktor's workflow

We present in this paper only the extraction system i.e. the first module of E-Gen, and its performance on this extraction and categorisation task. Section 2 shows a general system overview. In Section 3, we present the textual corpora and a short description of Aktor Interactive. In Section 4, we describe the algorithms used in the information extraction module. In Section 5, we show some results of our system before concluding and indicating future work.

## 2 System Overview

The main activity of Aktor Interactive is the processing of job offers on the Internet. As the Internet proposes new means for the recruitment market, Aktor

<sup>1</sup> <http://www.aktor.fr>

<sup>2</sup> <http://www.admincompta.fr> (Bookkeeper), <http://www.lesjeudis.com> (computing jobs), etc.

<sup>3</sup> <http://www.monster.com>, <http://www.cadremploi.fr>,  
<http://www.cadronline.com>

<sup>4</sup> <http://www.emploiregions.com> or <http://www.regionsjob.com>

is modifying its procedures to become able to integrate systems which carry out this processing as fast and judiciously as possible. An e-mail-box receives messages (sometimes with an attached file) containing the offer. After identification of the language, E-Gen parses the e-mail and examines attached file. Then the text containing the offer is extracted from the attachment. An external module, `wvWare`<sup>5</sup> processes MS-Word documents and produces a text document version as an output file, splitting this text into segments<sup>6</sup>. After filtering and lemmatisation, we are able to use a vectorial representation for each segment in order to assign a correct label to each segment using Support Vector Machines (SVM). This label sequence is processed by a corrective process which validates it or proposes a better sequence. At the end of the processing, an XML file is generated and sent to the Aktor Information system. The whole processing chain of E-Gen system is represented in figure 2.

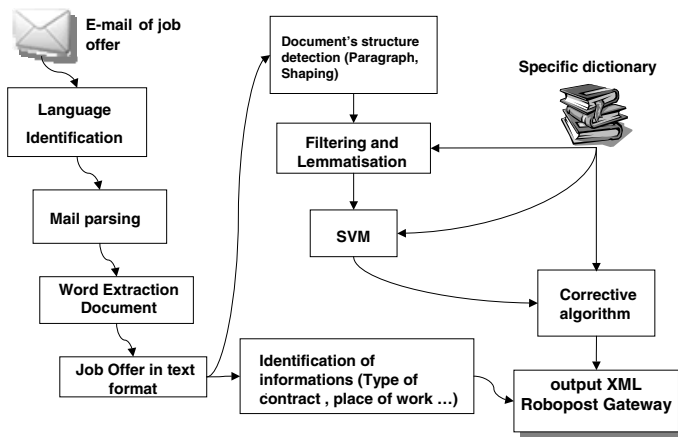


Fig. 2. E-Gen system overview

**Information extraction.** To post on-line a job vacancy, job boards require certain information. During the publication of job posting, some informations are required by the job board. So we need to find these fields in the job posting in order to include them in the XML document. We set up different solutions in order to locate each type of information:

- Salary: Regular expressions and rules were created to locate expressions such as "Salary: from X to Y", "Salary: between X and Y" or "X fixed salary with bonus", etc.
- Place of work: A table including area, city and department fields was created to find the location listing in a job posting. Most of the job boards categorise job postings according to the region to help job seekers in their job search.

<sup>5</sup> <http://wvware.sourceforge.net>

<sup>6</sup> In most situations, `wvWare` division matches the paragraph of the MSWord's document. Segmentation of text is a real issue, so we choose to use an existing tool.

- Company: In order to be able to integrate logos in job offers, a list of customers was plugged into the system to detect the company's name in the job postings.

Other information is recovered by similar processes (contract, reference, duration of mission, etc.). Finally, a report is sent to the user in order to show the fields correctly detected and the fields not found (either by extraction error or missing information in the job offer).

### 3 Corpora and Modelisation

#### 3.1 Corpora Description

We have selected a data subset from Aktor's database. This corpora is a mixture of different job listings in several languages. Initially we concentrated our study on French job posting because the French market represents the Aktor's main activity. This subset has been called the *Reference Corpora*. A job listing example is presented in table 1, translated from French to English (the content of the job offer is free, as we can see in this french example<sup>7</sup>, but it stays conventional as we find a rather similar presentation in every job listing and vocabulary according to every part as we shall see later on). The extraction of Aktor database made it possible to have an important corpora, without manual categorisation. A first analysis of this corpora shows that job offers often consist of similar blocks of information that remain, however, strongly unstructured. Each job posting is separated in four classes, as follow:

1. "Description\_of\_the\_company": Brief digest of the company that is recruiting.
2. "Title": presumably job title.
3. "Mission": a short job description.
4. "Profile": required skills and knowledge for the position. Contacts are generally included in this part.

---

<sup>7</sup> *Ce groupe français spécialisé dans la prestation d'analyses chimiques, recherche un: RESPONSABLE DE TRANSFERT LABORATOIRE. Sud Est. En charge du regroupement et du transfert d'activités de différents laboratoires d'analyses, vous étudiez, conduisez et mettez en oeuvre le sequencement de toutes les phases nécessaires à la réalisation de ce projet, dans le respect du budget previsionnel et des delais fixes. Vos solutions integrent l'ensemble des parametres de la demarche (social, logistique, infrastructures et materiels, informatique) et dessinent le fonctionnement du futur ensemble (Production, methodes et accreditations, developpement produit, commercial). De formation superieure Ecole d'ingenieur Chimiste (CPE) option chimie analytique environnementale, vous avez deja conduit un projet de transfert d'activite. La pratique de la langue anglaise est souhaitee. Merci d'adresser votre candidature sous la reference VA 11/06 par e-mail beatrice.lardon@atalan.fr.*



**Table 1.** Job postings example

This french firm, specialised in chemical analysis, is looking for:  
 PERSON IN CHARGE OF LABORATORY TRANSFER  
 South East  
 You will be in charge of regrouping the transfer activities of different analysis laboratories. You will analyse, conduct and implement the necessary phases of the project, respecting budgets and previously defined, dead lines.  
 Your solution will need to consider different parameters of the project (social, logistic, materials, data processing...) and integrate a roadmap (production, methods, accreditations, development, commercial... ).  
 Being a post graduate in chemical engineering with a focus on environmental analytical chemistry, you have already led an activity transfer project.  
 Fluent English required. Please send your CV and cover letter indicating reference number VA 11/06 to beatrice.lardon@atalan.fr

Table 2 shows a few statistics about our Reference Corpora.

**Table 2.** Corpora statistics

Number of job postings	D=1000	
Number total of Segments	P=15621	
Number of Segments "Title"	1000	6.34%
Number of Segments "Description of the company"	3966	25.38%
Number of Segments "Mission" description	4401	28.17%
Number of Segment "Profile" description	6263	40.09%

A pre-processing task of the corpora was performed to obtain a suitable representation in the Vector Space Model (VSM). Mainly deletion of the followings items : verbs and functional words (to be, to have, to be able to, to need,...), common expressions (for example, that is, each of,...), numbers (in numeric and/or textual format) and symbols such as \$, #, \*, etc. because these terms may introduce noise in the segment classification. Lemmatisation processing has also been performed to obtain an important reduction of the lexicon. It consists of finding the root of verbs and transform plural and/or feminine words to masculine singular form<sup>8</sup>. This process allows to decrease the curse of dimensionality [8] which raises severe problems of representing of the huge dimensions [9]. Other reduction mechanisms of the lexicon are also used: compound words are identified by a dictionary, then transformed into a unique term. All these processes allow us to obtain a representation in bag-of-words (a matrix of frequencies/absences of segment texts (rows) and a vocabulary of terms (columns)).

### 3.2 Markov's Machine

Preliminary experiments show that segment categorisation without segment positioning of a job posting is not enough and may be a source of errors. Figure 4

<sup>8</sup> So we can transform terms *sing*, *sang*, *sung*, *will sing* and possibly *singer* into *sing*.

shows that SVM produces a good classification of segments globally, but the job postings (documents) are rarely classified completely. Therefore due to the huge number of cases, the rules don't seem to be the best way to solve the problem. So we have implemented a machine with 6 states ("Start" (0), "Title" (1), "Description\_of\_the\_company" (2), "Mission" (3), "Profile" (4) and "End" (5)). Thus, we have considered each job posting as a succession of states in a Markov's machine. *Reference Corpora* has been analysed to determine the probabilities to switch from one state to another (transition). Matrix  $M$  (eq. 1) shows the values of the probabilities.

$$M = \begin{pmatrix} & \text{START} & \text{TITLE} & \text{DESCRIPTION} & \text{MISSION} & \text{PROFIL} & \text{END} \\ \text{START} & 0 & 0,01 & 0,99 & 0 & 0 & 0 \\ \text{TITLE} & 0 & 0,05 & 0,02 & 0,94 & 0 & 0 \\ \text{DESCRIPTION} & 0 & 0,35 & 0,64 & 0,01 & 0 & 0 \\ \text{MISSION} & 0 & 0 & 0 & 0,76 & 0,24 & 0 \\ \text{PROFIL} & 0 & 0 & 0 & 0 & 0,82 & 0,18 \\ \text{END} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (1)$$

Observing this matrix<sup>9</sup> allows us to learn a lot of things about the organisation of segments in a job posting. A job posting has a probability  $p = 0.99$  to start with a segment "Description" (2) while it is impossible that a job posting starts with a "Mission" or "Profile" segment (null probability). In the same way, each "Mission" segment can only be followed by another "Mission" or "Profil" segment. This matrix allows to build a Markov's machine shows in the figure 3.

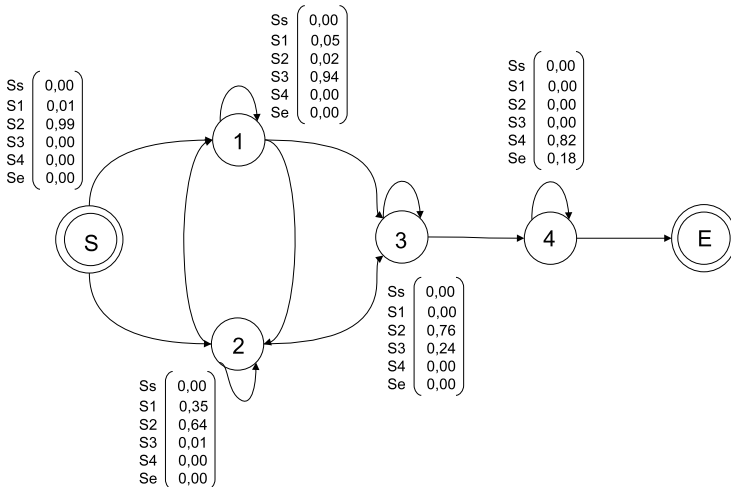


Fig. 3. Markov's machine used to correct wrong labels

<sup>9</sup> "Description" label corresponding in the matrix to "Description\_of\_the\_company".

## 4 Segment Categorisation Algorithms

### 4.1 Support Vector Machine Classification

SVM machines, proposed by Vapnik [10] have been successfully used in several tasks of machine learning. In particular, they offer a good estimation of the minimisation principle of the structural risk. The main ideas behind this method are:

- Data is mapped in a high dimensional space through a transformation based on a linear, polynomial or gaussian kernel.
- Classes are separated (in the new space) by linear classifiers, which maximise the margin (distance between the classes).
- Hyperplans can be determined by a few number of points: each of them is called a support vector.

Thus, the complexity of a classifier SVM depends, not on the dimension of the data space, but on the number of support vectors required to realise the best separation [11]. SVM has been already applied in the domain of the classification of the text in several works [12]. We have chosen to use SVM in this type of particular corpora as we have already seen good results in similar previous work [13]. We have used the implementation Libsvm [14] that allows to treat the multiclass problems in big dimensions.

### 4.2 Corrective Process

Our preliminary results obtained by the SVM method show a performant classification of segments. However, during the classification of a complete job posting, some segments were incorrectly classified, without regular behaviour (a "Description of the company" segment was detected in the middle of a paragraph profile; the last segment of the job posting was identified as a "Title", etc.). In order to avoid this kind of error, we applied a post-processing, based on the Viterbi algorithm [9,15]. The SVM classification for each segment provides a predicted class, and thus for a job posting, we have a class sequence (example: the sequence 0→2→2→1→3→3→4→5, i.e "Start" ↦ "Description of the company" ↦ "Description of the company" ↦ "Title" ↦ "Mission" ↦ "Mission" ↦ "Profile" ↦ End). A classical Viterbi algorithm will compute the probability of sequence. If the sequence is not probable, Viterbi's algorithm returns 0. When the sequence has a null probability our corrective process returns the sequence with a minimum error and maximal probability (compared to the original sequence generated by SVM).

First results were interesting but involved a considerable amount of processing time. We have introduced an improvement using Branch and Bound algorithm [16] for pruning the tree: once an initial solution is found, its error and probability are compared each time that a new sequence is processed. If the solution is not improved, the end of the sequence is not computed. The use of this algorithm enables us to reach an optimal solution, but not the best time (it have an exponential complexity). In test, this strategy computes sequences  $\leq 50$  symbols in approximately 4 seconds.

---

**Calcul next symbol()**

Processes the current sequence (Viterbi): full sequence, their probability, and the number of errors

**if** *the error of current sequence > max error found* **then**  
    return current sequence**end****if** *symbol is the last of the sequence* **then**    **if** *current error < max error* **then**  
        *maxerror = currenterror;*    **end**    return *sequence;***end****else**    **foreach** *symbol successor of the sequence* **do**        current sequence = **Calcul next symbol()**        **if** *current sequence is the best sequence* **then**            *bestsequence = currentsequence;*            **if** *current error < max error* **then**                *maxerror = currenterror;*            **end**        **end**    **end****end**

---

**Algorithm 1.** Corrective Process algorithm with Branch and Bound method

## 5 Results and Discussion

We have used a corpora of  $D = 1,000$  job postings split into  $P = 15,621$  segments. Each test was carried out 20 times with a random distribution between the test corpora and the training corpora.

Figure 4 shows the comparison between the results obtained by the Support Vector Machines and the corrective process. The curves present the number of segments unrecognized according to the size of the training corpora. On the left, we present the results of SVM machines alone (dotted line) applied to the segment classification task. The baseline is computed with the most probably label class, i.e. "Profile" (cf. table 2). The results are good and show that even with a small fraction of learning patterns (20% of total), the SVM classifier obtains a low rate of misclassification (less than 10% error). The corrective process (solid line) always gives better results than SVM whatever fraction of patterns in learning.

The curve on the right hand of the figure 4 compares the results obtained by each method according to unrecognized job postings. We can also see a considerable improvement of the number of job postings recognized with the corrective process. SVM algorithm reaches a maximum of  $\approx 50\%$  of unrecognized job postings while the corrective process gives 20% of unrecognized job postings, so an improvement of more than 50% on the SVM score.

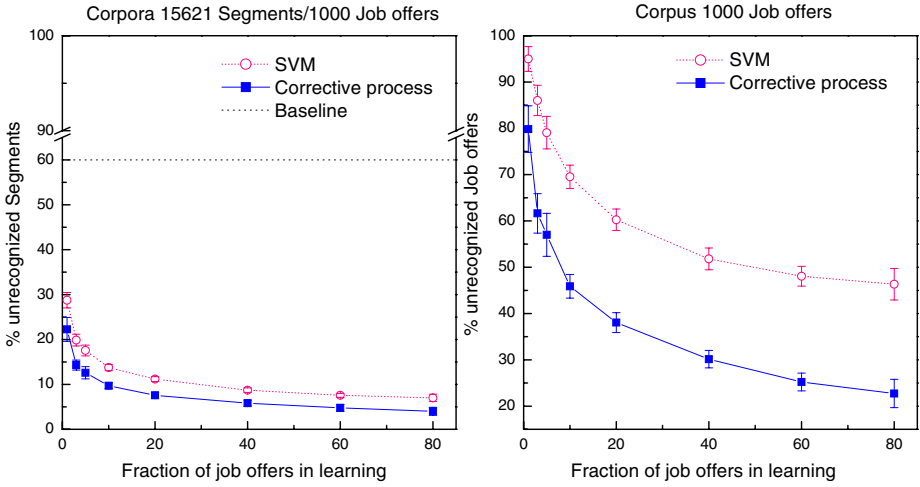


Fig. 4. Results of SVM and corrective process for segments on the left and for jobs offers on the right



Fig. 5. Block boundary errors

An analysis of wrongly-classified job postings, shows that about 10% of the job postings contains one or two errors. These misclassified segments generally correspond block boundaries between 2 different categories [17], as it is shown in figure 5. So, the obtained sequence, for example 1 (cf. 3.1) is 0→2→1→3→3→3→4→4→5 but the correct sequence is 0→2→1→3→3→4→4→4→5. The wrong segment is reproduced in table 3.

We observed that some important terms are present in two different categories, leading to a wrong classification. In particular, in this example we are talking about terms like **project**, **activity transfer** that corresponding to "Mission" and "Profile" categories. The segment is classified as "Profile". In fact, this segment occurs at the boundary between the "Mission" and "Profile" segments, and the sequence is probable (Viterby’s probability is not null), so this error is not

Table 3. Example of a misclassified segment

<i>De formation superieure Ecole d’ingenieur Chimiste (CPE) option chimie analytique environnementale, vous avez deja conduit un <b>projet de transfert d’activite</b>.</i>
Translation of the wrong classified segment: Being a post graduate in chemical engineering with a focus on analystic environmental chemistry, you have already lead a <b>project of transfer of activity</b> .

corrected by the corrective process. The improvement of the block boundary's detection is one of the ways that we are currently exploring [18] in order to increase the performance of our system.

## 6 Conclusion and Future Work

Processing job posting information is a difficult task because the information flow is still strongly unstructured. In this paper we show the categorisation module, the first component of E-Gen, a modular system to treat jobs listings automatically. The first results obtained through SVM were interesting (approximately 10% error for a training corpora of 80%). The application of the corrective process improves the results by approximately 50% on the SVM score and considerably decreases errors such as "wrongly classified segments that are isolated" with very good computing times. Informations such as salary (minimum, maximum salary and currency), place of work and categorisation of the occupation are correctly detected to send pertinent information about the job posting to the job boards. The first module of E-Gen is currently in test on Aktor's server and offers a considerable time saving in the daily treatment of job offers. E-Gen is a independent and portable database, because it is a modular system with e-mail as input and XML documents as output. The promising results in this paper allow us to continue the E-Gen project with the relevance ranking of candidate responses. Several approaches (information retrieval, machine learning, automatic summarisation) will be considered to resolve these problems with a minimal cost in terms of human intervention.

## Acknowledgement

Autors thanks to Jean Arzalier, Eric Blaudez, ANRT (*Agence Nationale de la Recherche Technologique*, France) and Aktor Interactive that partially supported this work (Grant Number CIFRE 172/2005).

## References

1. Bizer, C., Heese, R., Mochol, M., Oldakowski, R., Tolksdorf, R., Eckstein, R.: The impact of semantic web technologies on job recruitment processes. In: International Conference Wirtschaftsinformatik (WI 2005), Bamberg, Germany (2005)
2. Rafter, R., Bradley, K., Smyt, B.: Automated Collaborative Filtering Applications for Online Recruitment Services, 363–368 (2000)
3. Rafter, R., Smyth, B.: (Passive Profiling from Server Logs in an Online Recruitment Environment)
4. Bourse, M., Leclère, M., Morin, E., Trichet, F.: Human resource management and semantic web technologies. In: Proceedings, 1st International Conference on Information & Communication Technologies: from Theory to Applications (ICTTA) (2004)

5. Morin, E., Leclère, M., Trichet, F.: The semantic web in e-recruitment. In: The First European Symposium of Semantic Web (ESWS'2004) (2004)
6. Rafter, R., Smyth, B., Bradley, K.: (Inferring Relevance Feedback from Server Logs: A Case Study in Online Recruitment)
7. Zighed, D.A., J., C.: Data Mining and CV analysis 17, 189–200 (2003)
8. Bellman, R.: Adaptive Control Processes. Princeton University Press, Princeton (1961)
9. Manning, D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (2002)
10. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
11. Joachims, T.: Making large scale SVM learning practical. Advances in kernel methods: support vector learning, pp. 169–184. MIT Press, Cambridge (1999)
12. Grilheres, B., Brunessaux, S., Leray, P.: Combining classifiers for harmful document filtering. In: RIAO 2004, pp. 173–185 (2004)
13. Kessler, R., Torres-Moreno, J.M., El-Bèze, M.: Classification automatique de courriers électroniques par des méthodes mixtes d'apprentissage, 93–112 (2006)
14. Fan, R.-E., Chen, P.-H., Lin, C.-J.: Towards a Hybrid Abstract Generation System. In: Working set selection using the second order information for training SVM, pp. 1889–1918 (2005)
15. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimal decoding algorithm 13, 260–269 (1967)
16. Land, A.H., Doig, A.G.: An Automatic Method of Solving Discrete Programming Problems 28, 497–520 (1960)
17. El-Bèze, M., Torres-Moreno, J., Béchet, F.: Un duel probabiliste pour départager deux Présidents. In: RNTI coming soon, pp. 1889–1918 (2007)
18. Reynar, J., Ratnaparkhi, A.: A Maximum Entropy Approach to Identifying Sentence Boundaries. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, Washington, D.C., pp. 16–19 (1997)

# How Context and Semantic Information Can Help a Machine Learning System?

Sonia Vázquez, Zornitsa Kozareva, and Andrés Montoyo

Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Alicante  
{svazquez,zkozareva,montoyo}@dlsi.ua.es

**Abstract.** In Natural Language Processing there are different problems to solve: lexical ambiguity, summarization, information extraction, speech processing, etc. In particular, lexical ambiguity is a difficult task that nowadays is still open to new approaches. In fact, there is still a lack of systems that solve efficiently this kind of problem. At present, we find two different approaches: knowledge systems and machine learning systems. Recent studies demonstrate that machine learning systems obtain better results than knowledge systems but there is a problem: the lack of annotated contexts and corpus to train the systems. In this work, we try to avoid this situation by combining a new machine learning system with a knowledge based system.

## 1 Introduction

Word Sense Disambiguation (WSD) is an open research field in Natural Language Processing (NLP). The task of WSD consists in assigning the correct sense to words in a particular context using an electronic dictionary as the source of words definitions. This is a difficult problem that is receiving a great deal of attention from the research community.

Designing a system for Natural Language Processing (NLP) requires a large knowledge on language structure, morphology, syntax, semantics and pragmatic nuances. All of these different linguistic knowledge forms, however, have a common associated problem, their many ambiguities, which are difficult to resolve. In fact, there is a study that demonstrates that any system has an adequate accuracy for all words [8].

Many systems use the lexical resource WordNet. But it is not a perfect resource for word-sense disambiguation, because it has the problem of the fine granularity of WordNet's sense distinctions [6]. This problem causes difficulties in the performance of automatic word-sense disambiguation with free-running texts. Several authors have stated that the divisions of a proposed sense in the dictionary are too fine for Natural Language Processing. To solve this problem [5] have developed a new resource to reduce the fine granularity of senses. This resource is named WordNet Domains. Applying domains to WSD contributes with a relevant information to establish semantic relations between word senses. For example, "bank" has ten senses in WordNet but three of them "bank#1",



”bank #3” and ”bank #6” are grouped into the same domain label ”Economy”, whereas ”bank#2” and ”bank#7” are grouped into domains labels ”Geography” and ”Geology”.

A proposal in WSD using domains has been developed in [1]; they use WordNet Domains as lexical resource, but from our point of view they don’t make good use of glosses information.

In this paper we present a new approximation for the resolution of the lexical ambiguity that appears when a given word in a context has several different meanings. The resolution of the lexical ambiguity can help other NLP applications such as Machine Translation (MT), Information Retrieval (IR), Text Processing, Grammatical Analysis, Information Extraction, hypertext navigation and so on. Our intention is to combine a machine learning system with different modules of knowledge based systems. First of all we have developed different features based on the information provided by context, such as: 3 words before target word, 3 words next, part of speech of 3 words before, first name before target word, etc. Once we have obtained these features we have introduced as new features different modules which are based on knowledge based systems. These modules are based on information provided by the resource of Relevant Domains, and the information provided by the Latent Semantic Analysis module (LSA) [2]. Moreover, we have introduced the module of WordNet Similarity [7] which measures the similarity between a pair of words: noun-noun, noun-verb, noun-adjective, etc. With all this new information that not depends on training texts and corpus we hope to improve the results of the initial system.

The organization of this paper is: after this introduction, in section 2 we describe the new lexical resource, named Relevant Domains. In section 3, the LSA module is described. In section 4 the new WSD method is presented combining all the information. And finally sections 5 and 6 show the discussion and the final conclusions.

## 2 WordNet Domains Description

The semantic domains provide a natural way to establish the semantic relations among words. They can be used to describe texts and to assign a specific domain from previously established domain hierarchy.

Based on this idea, a new resource called WordNet Domains (WND) [5] has been developed. This resource uses information from WordNet [6] and labels each word sense with semantic domains from a set of 200 domain labels. These labels are obtained from the Dewey Decimal Classification and are hierarchically organized. This information is complementary to WordNet and is useful to obtain new relations among the words.

In order to obtain the WND resource, each word of WordNet is annotated with one or more domain labels. One of the most important characteristic of WND is that each domain label can be associated to different syntactic categories. This is an interesting feature because we can relate words of different syntactic

categories with the same domain and obtain new relations that previously does not exist in WordNet.

For example, the domain ‘Economy’ is associated with the nouns (bank, money, account, banker, etc), the verbs (absorb, amortize, discount, pay, etc) and the adjectives (accumulated, additional, economic, etc). Moreover, these domain labels have been associated to different senses of the same word and thus we can distinguish the meaning of each word using the domains. The word “plant” has three different domain senses: ‘Industry, Botany, Theatre’ and in order to establish its word sense, we can use the domain information of other words that are seen in the context of “plant” (“it is an industrial plant to manufacture automobiles”, “a plant is a living organism lacking the power of locomotion”).

Taking advantage of the properties of this resource, we formulate the following hypothesis: the conceptual representation of a text can be obtained when the contextual information provided by its words is used.

In WordNet, each word sense has a definition<sup>1</sup> like in a dictionary and the words in the gloss are used to obtain the specific context for the sense. Respectively, the word sense has a domain label which contains the global concept for this sense. Our assumption is that words that form part of the gloss are highly probable to be associated to the same concept of the word. For instance, “plant#1”<sup>2</sup> is associated to the domain ‘Industry’. Its gloss contains: ‘buildings for carrying on industrial labor; ”they built a large plant to manufacture automobiles”’. From the gloss, the words “building”, “carry”, “industrial”, “labor”, “plant”, “manufacture” and “automobile” are semantically related to the domain ‘Industry’ and thus they can help us to understand the concept of the definition word.

Taking into account this principle, we extracted from WordNet a list with all words and their associated domains. Then, we used the information provided by the context to build the conceptual representation space of LSA.

### 3 Latent Semantic Analysis

The traditional usage of LSA is based on a text corpus represented as a  $M \times N$  co-occurrence matrix, where the rows  $M$  are words and the columns  $N$  are documents, phrases, sentences or paragraphs. Each cell in this matrix contains the number of times that a word occurs in the context of a column.

Once the matrix is obtained, it is decomposed using Singular Value Decomposition (SVD). In this way the initial dimensions are reduced into a new distribution which is based on similar contexts. This reduction makes the similarity among the words and the contexts to become more apparent.

Our approach is based on the idea that semantically related words appear in the same contexts. However, the contexts we use are not a specific corpus divided in documents or paragraphs, but words related to a specific concept (e.g. domain) that belongs to a predefined hierarchy. In our case, the domains

<sup>1</sup> Gloss.

<sup>2</sup> Plant with sense one.

are derived from WND and with this information we construct the conceptual matrix of LSA.

However, we want to rank the words not only on the basis of their meaning, but also on the basis of their co-occurrences with other words. Therefore, we applied the Mutual Information (MI) (1) and Association Ratio (AR) (2) measures which can relate the words with the domains.

$$MI(w_1, w_2) = \log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \quad (1)$$

$$AR(w, Dom) = Pr(w|Dom) \log_2 \frac{Pr(w|Dom)}{Pr(w)} \quad (2)$$

MI provides information about the pairwise probability of two words  $w_1$  and  $w_2$  compared to their individual probabilities. When there is a real association between two words  $w_1$  and  $w_2$ , their joint probability  $P(w_1, w_2)$  is much larger than  $P(w_1)P(w_2)$ , and  $MI(w_1, w_2) \gg 0$ . For the cases where  $w_1$  and  $w_2$  are not related,  $P(w_1, w_2) \approx P(w_1)P(w_2)$ , therefore  $MI(w_1, w_2) \approx 0$ . When  $w_1$  and  $w_2$  are in complementary distribution, then  $P(w_1, w_2)$  is less than  $P(w_1)P(w_2)$ , and  $MI(w_1, w_2) \ll 0$ .

Adapting this notion to our approach, we used  $w_1$  in the aspect of a word we are observing and  $w_2$  in the aspect of a domain  $D$  from the WND that corresponds to the word  $w_1$ . The values are normalized with the number of word–domain pairs  $N$  in the WND. Once the relation between the words and the domains is obtained, AR is applied and the conceptual space of LSA is constructed.

This method has been applied to identify semantic variability expressions obtaining promising results [4].

## 4 Selected Features

Once the knowledge based modules have been developed, we can start to explain how we can introduce this information in our machine learning system. We have selected a set of different features to build our system. These features have been selected from the information provided by context and from external resources in order to take advantage of different options. In our case, we have used information provided by WordNet Domains, WordNet Similarity module and Latent Semantic Analysis module.

Next we can see the set of the 20 different features we have selected to our system:

- F1: Target word, part of speech:  $w_0$ , {NVAR...}
- F2: Three words before:  $w_{-1}$ ,  $w_{-2}$ ,  $w_{-3}$
- F3: Three words next:  $w_{+1}$ ,  $w_{+2}$ ,  $w_{+3}$
- F4: Part of speech 3 words before: {NVAR...}, {NVAR...}, {NVAR...}
- F5: Part of speech 3 words next: {NVAR...}, {NVAR...}, {NVAR...}

- F6: First name before target word:  $w_{bn}$
- F7: First name next target word:  $w_{nn}$
- F8: First verb before target word:  $w_{bv}$
- F9: First verb next target word:  $w_{nv}$
- F10: Three relevant domains of the target word:  $domain_1, domain_2, domain_3$
- F11: Cosine value of each sense of target word:  $cos_1, cos_2, \dots$
- F12: LSA module:  $LSA_{dom1}, LSA_{dom2}, \dots$
- F13: Bigram words before target word:  $w_{-1,-2}$
- F14: Bigram words next target word:  $w_{+1,+2}$
- F15: Bigram domains before target word:  $d_{-1,-2}$
- F16: Bigram domains next target word:  $d_{+1,+2}$
- F17: WordNet Similarity value (WNS): first name before and  $w_0$
- F18: WordNet Similarity value (WNS): first name next and  $w_0$
- F19: WordNet Similarity value (WNS): first verb before and  $w_0$
- F20: WordNet Similarity value (WNS): first verb next and  $w_0$

In this set of features we can find typical features such as: F1 to F9. In this case, these features refer to the context surrounding the target word. But from F10 to F21, we find different features extracted from external resources. In next subsections we explain each kind of feature.

#### 4.1 F10: Three Relevant Domains of the Target Word

This feature uses the WordNet Domains resource in order to extract the three more relevant domains of the target word. In this case, we use the Association Ratio (AR) measure to extract the relevant domains of each word. This measure has been applied before to WSD [10].

The AR measure obtains the most common and relevant domains to each word of WordNet and assigns to each pair of word-domain a value from 0 to 1. Values around 1 indicate that the pair word-domain are semantically closer than other pairs with values around 0.

Table 1 shows the relevant domains for word "plant" from WordNet Domains.

As we can see, the three more relevant domains to plant are: agriculture, botany and biology. These domains are extracted from the number of times the word plant appears in WordNet with each domain. We have selected only the WordNet context because it is obtained from the definitions of each sense and it is not based in a particular field, topic, etc.

#### 4.2 F11: Cosine Value of Each Sense of Target Word

This feature extracts the cosine value for each possible sense of the target word. In this case, we try to evaluate which is the most appropriate sense according to the context the word appears. To calculate cosine we need to establish the context surrounding the target word and which are the coordinates.

In our case, the coordinates are the different domains a word pertains and the context words are those of the sentence the target word appears.

**Table 1.** Relevant Domains

Word	Domain	AR
plant	agriculture	0.102860
plant	botany	0.071716
plant	biology	0.064123
plant	entomology	0.022920
plant	archaeology	0.019603
plant	mountaineering	0.019178
plant	alimentation	0.010787
plant	ecology	0.006275
plant	industry	0.003025
plant	building_industry	0.002533
plant	pharmacy	0.002441
plant	physiology	0.002435
plant	anatomy	0.002379
plant	medicine	0.002247
plant	architecture	0.002211

To obtain the cosine value we build two types of vectors: context vector and sense vector. Context vector is obtained from words in the context sentence. Sense vectors are obtained from the information of glosses of each sense of the target word. So, for each pair of context-sense vectors we obtain the cosine value between 0-1. In the same way as AR, the cosine measure closer to 1 indicates that the sense is more accurate to the context the target word appears.

### 4.3 F12: LSA over Each Sense

This feature exploits the information provided from the context the target word appears. With LSA we can obtain how much similar is a word sense with respect to the context. In this case, we build a matrix with the 164 domains of WordNet Domains. This matrix is obtained from the information of WordNet glosses. So, from a sentence or a context of  $x$  words, we can decide which are the most similar domain contexts. For example, if we want to disambiguate the word "*bank*" in the sentence: "Everyone has a bank account and a credit card". We use the LSA module with the domain matrix and the result will be domains as: Economy, commerce, industry, etc.

So, in our case, we take a context of 20 words surrounding the target word and obtain the result of the LSA module (those domains which similarity to the context is bigger than 0.8).

### 4.4 Bigram Domains

This feature combines the two more relevant domains of the previous words and the next words to the target word. This information is useful to identify which is the more appropriate context according to domain information. Each time

we have referred to domain information is only for nouns, verbs, adjectives or adverbs.

#### 4.5 WordNet Similarity Value

The module of WordNet Similarity [7], extracts the similarity between different word pairs. This module can obtain the more appropriate sense using different relatedness measures. In our case, we use the Lin and JCN measures.

### 5 Discussion

This system is our first approach to solve the lexical ambiguity. We know that there are new lexical resources that can help us to obtain better results. This is the case of the Extended WordNet [9] where each word of each gloss is annotated with its correct sense. So, in this case, we can obtain a better approximation of Relevant Domains if we use the information of this resource. This is our next step after test our system.

Also, there is another annotation of WordNet using the information provided by SUMO [3]. In this case, SUMO ontology has a more specialized set of labels associated to the senses of WordNet.

Our intention is to develop this preliminary system and once obtained the results study the necessity of improving each module to obtain better results. We are using the test of SEMEVAL English all words task to evaluate our system.

### 6 Conclusions

We have described an approach for WSD using an hybrid system that combines a machine learning system with different modules based on knowledge based systems. Our hypothesis is that domains constitute a fundamental feature of text coherence, therefore words occurring in coherent portion of texts maximize the domain similarity. This information is useful to determine which is the correct sense of a word. Moreover, the context where words appear can help us to determine whether a word has a sense or another. This approximation is implemented using the LSA module classifying words into domain labels instead of classifying words into documents or paragraphs as traditionally.

As future work, we have pending the evaluation of our system with the test of the English all words task of SEMEVAL. And once obtained the results we are going to study the effect of adding new information to improve this previous modules. We will use the information provided by Extended WordNet and the information provided by SUMO.

### Acknowledgements

This research has been partially supported by the framework of the project QALL-ME (FP6-IST-033860), which is a 6th Framenwork Research Programme

of the European Union (EU), and by the Spanish Government, project TEXT-MESS (TIN-2006-15265-C06-01).

## References

1. Magnini, B., Strapparava, C.: Experiments in word domain disambiguation for parallel texts. In: Proceedings of SIGLEX. Workshop on Word Senses and Multilinguality (2000)
2. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. *JASIS* 41(6), 391–407 (1990)
3. Webster, J.J., Chow, I.C.: Mapping framenet and sumo with wordnet verb: Statistical distribution of lexical-ontological realization. In: Gelbukh, A., Reyes-Garcia, C.A. (eds.) MICAI 2006. LNCS (LNAI), vol. 4293, pp. 262–268. Springer, Heidelberg (2006)
4. Kozareva, Z., Vázquez, S., Montoyo, A.: The usefulness of conceptual representation for the identification of semantic variability expressions. In: CICLing, pp. 325–336 (2007)
5. Magnini, B., Cavaglià, G.: Integrating subject field codes into wordnet. In: Gavriliadou, M., Crayannis, G., Markantonatu, S., Piperidis, S., Stainhaouer, G. (eds.) Second International Conference on Language Resources Proceedings of LREC-2000 and Greece Evaluation, Athens, pp. 1413–1418 (2000)
6. Ide, N., Veronis, J.: Introduction to the special issue on word sense disambiguation: The state of the art. In: Computational Linguistics, pp. 1–40 (1998)
7. Pedersen, T., Patwardhan, S., Michelizzi, J.: Wordnet:similarity - measuring the relatedness of concepts. In: Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-2004), pp. 1024–1025 (2004)
8. Saarikoski, H.M.T., Legrand, S., Gelbukh, A.F.: Defining classifier regions for wsd ensembles using word space features. In: MICAI, pp. 855–867 (2006)
9. Moldovan, D.I., Harabagiu, S.M., Miller, G.A.: Wordnet 2 - a morphologically and semantically enhanced resource. In: SIGLEX
10. Vázquez, S., Montoyo, A., Rigau, G.: Using relevant domains resource for word sense disambiguation. In: IC-AI, pp. 784–789 (2004)

# Auditory Cortical Representations of Speech Signals for Phoneme Classification

Hugo L. Rufiner<sup>1,2</sup>, César E. Martínez<sup>1,2</sup>, Diego H. Milone<sup>2</sup>,  
and John Goddard<sup>3</sup>

<sup>1</sup> Laboratorio de Señales e INteligencia Computacional (SINC), Depto Informática  
Facultad de Ingeniería y Cs Hídricas - Universidad Nacional del Litoral  
CC 217, Ciudad Universitaria, Paraje El Pozo, S3000 Santa Fe, Argentina  
Tel.: +54 (342) 457-5233 x 148  
[lrufiner@fich.unl.edu.ar](mailto:lrufiner@fich.unl.edu.ar)

<sup>2</sup> Facultad de Ingeniería, Universidad Nacional de Entre Ríos, Argentina

<sup>3</sup> Dpto. de Ingeniería Eléctrica - UAM-Iztapalapa, México

**Abstract.** The use of biologically inspired, feature extraction methods has improved the performance of artificial systems that try to emulate some aspect of human communication. Recent techniques, such as independent component analysis and sparse representations, have made it possible to undertake speech signal analysis using features similar to the ones found experimentally at the primary auditory cortex level. In this work, a new type of speech signal representation, based on the spectro-temporal receptive fields, is presented, and a problem of phoneme classification is tackled for the first time using this representation. The results obtained are compared, and found to greatly improve both an early auditory representation and the classical front-end based on Mel frequency cepstral coefficients.

## 1 Introduction

The development of new techniques in the signal analysis and representation fields promises to overcome some of the limitations of classical methods for solving real problems with complex signals, such as those related to human speech recognition. Furthermore, novel techniques for signal representation, for example those using overcomplete dictionaries, provide an important new way of thinking about alternative solutions to problems such as denoising or automatic speech recognition. Significant connections have been found between the way the brain processes sensory signals and some of the principles that support these new approaches [1,2].

In the process of human communication, the inner ear –at the cochlea level– carries out a complex time-frequency analysis and encodes a series of meaningful cues in the discharge patterns of the auditory nerve. These early auditory representations, or *auditory spectrograms*, have been widely studied and mathematical and computational models have been developed that allow them to be estimated approximately [3].



In spite of the knowledge available about early auditory representations, the principles that support speech signal representation at higher sensorial levels, as in the primary auditory cortex, are still the object of research [4]. Among these principles two can be singled out, namely, the need for very few active elements in a signal's representation, and the statistical independence between these elements. Using these principles, it is then possible to establish a model for cortical representations that displays important correlations with experimentally obtained characteristics from their physiological counterparts [5,6].

In order to obtain this cortical model, techniques related to *independent component analysis* (ICA) and *sparse representations* (SR) are used [7,8]. These techniques can emulate the behavior of cortical neurons using the notion of *spectro-temporal receptive fields* (STRF) [9]. STRF can be defined as the required optimal stimulus so that an auditory cortical neuron responds with the largest possible activation. Different methods, such as inverse correlation, are used to estimate them from mammal neuronal activity data [10].

In this work, by making use of the time-frequency representations of the auditory spectrograms of speech signals, a dictionary of two-dimensional optimal atoms is estimated. Based on this STRF dictionary, a sparse representation that emulates the cortical activation is computed. This representation is then applied to a phoneme classification task, designed to evaluate the representation's suitability.

This work is organized as follows: Section 2 presents the method for the speech signal representation that is used in the paper. In particular, 2.3 explains how this representation can include one involving the primary auditory cortex. Section 3 details the data used in the phoneme classification experiments as well as the steps used to obtain the cortical representation patterns. Section 4 presents the experimental results together with a discussion. Finally, Section 5 summarizes the contributions of the present paper and outlines future research.

## 2 Sparse and Factorial Representations

### 2.1 Representations Based on Discrete Dictionaries

There are different ways of representing a signal using general discrete dictionaries. For the case where the dictionary forms a unitary or orthogonal transformation, the techniques are particularly simple. This is because, among other aspects, the representation is unique. However, in the general, non-orthogonal case, a signal can have many different representations using a single dictionary. In these cases, it is possible to find a suitable representation if additional criteria are imposed. For our problem, these criteria can be motivated by obtaining a representation with sensorial characteristics which are sparse and independent [11], as mentioned in the introduction. Furthermore, it is possible to find an optimal dictionary using these criteria [12].

A sparse code is one which represents the information in terms of a small number of descriptors taken from a large set [12]. This means that a small fraction of the elements from the code are used actively to represent a typical

pattern. In numerical terms, this signifies that the majority of the elements are zero, or ‘almost’ zero, most of the time [13,14].

It is possible to define measures or norms that allow us to quantify how sparse a representation is; one way is using either the  $\ell_0$  or the  $\ell_1$  norms. An alternative way is to use a probability distribution. In general one uses a distribution with a large positive kurtosis. This results in a distribution with a large thin peak at the origin and long tails on either side. One such distribution is the Laplacian. In the statistical context it is relatively simple to include aspects related to the independence of the coefficients, which connect this approach with ICA [7].

In the following subsection a formal description is given of a statistical method which estimates an optimal dictionary and the corresponding representation [1].

### 2.2 Optimal Sparse and Factorial Representations

Let  $\mathbf{x} \in \mathbb{R}^N$  be a signal to represent in terms of a *dictionary*  $\Phi$ , with size  $N \times M$ , and a set of coefficients  $\mathbf{a} \in \mathbb{R}^M$ . In this way, the signal is described as:

$$\mathbf{x} = \sum_{\gamma \in \Gamma} \phi_{\gamma} a_{\gamma} + \varepsilon = \Phi \mathbf{a} + \varepsilon , \tag{1}$$

where  $\varepsilon \in \mathbb{R}^N$  is the term for additive noise and  $M \geq N$ . The dictionary  $\Phi$  is composed of a collection of waveforms or parameterized functions  $(\phi_{\gamma})_{\gamma \in \Gamma}$ , where each waveform  $\phi_{\gamma}$  is an *atom* of the representation.

Although (1) appears very simple, the main problem is that for the most general case  $\Phi$ ,  $\mathbf{a}$  and  $\varepsilon$  are unknown, thus there can be an infinite number of possible solutions. Furthermore, in the noiseless case (when  $\varepsilon = \mathbf{0}$ ) and given  $\Phi$ , if there are more atoms than there are samples of  $\mathbf{x}$ , or if the atoms don’t form a base, then non-unique representations of the signal are possible. Therefore, an approach that allows us to select one of these representations has to be found. In this case –although this is a linear system– the coefficients chosen to be part of the solution generally form a non-linear relation with the data  $\mathbf{x}$ . For the complete, noiseless case the relationship between the data and the coefficients is linear and it is given by  $\Phi^{-1}$ . For classical transformations, such as the discrete Fourier transform, this inverse is simplified because  $\Phi^{-1} = \Phi^*$  (with  $\Phi \in \mathbb{C}^{N \times M}$  and  $\Phi^*(i, j) = \overline{\Phi(j, i)}$ ).

When  $\Phi$  and  $\mathbf{x}$  are known, an interesting way to choose the set of coefficients  $\mathbf{a}$  from among all the possible representations, consists in finding those  $a_i$  which make the representation as sparse and independent as possible. In order to obtain a sparse representation, a distribution with positive kurtosis can be assumed for each coefficient  $a_i$ . Further, assuming the statistical independence of the  $a_i$ , a joint *a priori* distribution satisfies:

$$P(\mathbf{a}) = \prod_i P(a_i) . \tag{2}$$

---

<sup>1</sup> Although two-dimensional patterns are used, for clearness we only describe the one-dimensional case.

The system appearing in (II) can also be seen as a generative model. Following the customary terminology used in the ICA field, this means that signal  $\mathbf{x} \in \mathbb{R}^N$  is generated from a set of sources  $a_i$  (in the form of a state vector  $\mathbf{a} \in \mathbb{R}^M$ ) using a mixture matrix  $\Phi$  (of size  $N \times M$ , with  $M \geq N$ ), and including an additive noise term  $\varepsilon$  (Gaussian, in most cases).

The state vector  $\mathbf{a}$  can be estimated from the *posterior* distribution:

$$P(\mathbf{a}|\Phi, \mathbf{x}) = \frac{P(\mathbf{x}|\Phi, \mathbf{a})P(\mathbf{a})}{P(\mathbf{x}|\Phi)} . \tag{3}$$

Thus, a *maximum a posteriori* estimation of  $\mathbf{a}$  would be:

$$\hat{\mathbf{a}} = \arg \max_{\mathbf{a}} [\log P(\mathbf{x}|\Phi, \mathbf{a}) + \log P(\mathbf{a})] . \tag{4}$$

When  $P(\mathbf{a}|\Phi, \mathbf{x})$  is sufficiently smooth, the maximum can be found by the method of gradient ascent. The solution depends on the functional forms assigned to the distributions for the noise and the coefficients, giving rise to different methods for finding the coefficients. Lewicki and Olshausen [15] proposed the use of a Laplacian *a priori* distribution with parameter  $\beta_i$ :

$$P(a_i) = \alpha \exp(-\beta_i |a_i|) , \tag{5}$$

where  $\alpha$  is a normalization constant. This distribution, with the assumption of Gaussian additive noise  $\varepsilon$ , results in the following updating rule for  $\mathbf{a}$ :

$$\Delta \mathbf{a} = \Phi^T \Lambda_\varepsilon \varepsilon - \beta^T |\mathbf{a}| , \tag{6}$$

where  $\Lambda_\varepsilon$  is the inverse of the noise covariance matrix  $\mathcal{E}[\varepsilon^T \varepsilon]$ , with  $\mathcal{E}[\cdot]$  denoting the expected value.

To estimate the value of  $\Phi$ , the following objective function can be maximized [15]:

$$\hat{\Phi} = \arg \max_{\Phi} [\mathcal{L}(\mathbf{x}, \Phi)] , \tag{7}$$

where  $\mathcal{L} = \mathcal{E}[\log P(\mathbf{x}|\Phi)]_{P(\mathbf{x})}$  is the likelihood of the data. This likelihood can be found by marginalizing the following product of the conditional distribution of the data, given the dictionary and the coefficients, together with the coefficients *a priori* distribution:

$$P(\mathbf{x}|\Phi) = \int_{\mathbb{R}^M} P(\mathbf{x}|\Phi, \mathbf{a})P(\mathbf{a}) d\mathbf{a} , \tag{8}$$

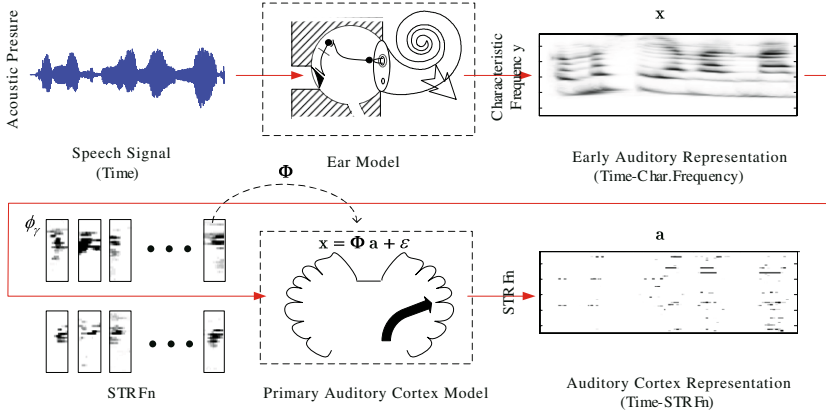
where the integral is over the  $M$ -dimensional state space of  $\mathbf{a}$ .

The objective function in (7) can be maximized using gradient ascent with the following update rule for the matrix  $\Phi$  [16]:

$$\Delta \Phi = \eta \Lambda_\varepsilon \mathcal{E}[\varepsilon \mathbf{a}^T]_{P(\mathbf{a}|\Phi, \mathbf{x})} , \tag{9}$$

where  $\eta$ , in the range  $(0, 1)$ , is the learning rate.

In this iterative way, the dictionary  $\Phi$  and the coefficients  $\mathbf{a}$  were obtained.



**Fig. 1.** Schematic diagram of the method used for estimating the auditory cortical representation

### 2.3 Auditory Cortical Representations

The properties of sensorial systems should coincide with the statistics of their perceived stimuli [17]. If a simple model of these stimuli is assumed, as the one outlined in (1), it is possible to estimate their properties from the statistical approach presented in the previous section.

The early auditory system codes important cues for phonetic discrimination, such as the ones found in the auditory spectrograms. In these representations –of a higher level than the acoustic one– some non-relevant aspects of the temporal variability of the sound pressure signal that arrives at the eardrum have been eliminated. Among these superfluous aspects, for example, is the relative phase from some acoustic waveforms [18]. Hence, following this biological simile, this representation forms a good starting point to attain more complex ones.

The obtention of a dictionary of two-dimensional atoms  $\Phi$ , corresponding to time-frequency features estimated from data  $\mathbf{x}$  of the auditory spectrogram, is equivalent to the STRF of a group of cortical neurons. Therefore, the activation level of each neuron can be assimilated with the coefficients  $a_\gamma$  in (1). Figure 1 shows a schematic diagram of the method adopted for estimating the cortical representation.

Kording *et al* carried out a qualitative analysis of dictionaries obtained in a similar way, and they found that their properties compared favorably with those of the natural receptive fields [5].

## 3 Experiments and Data

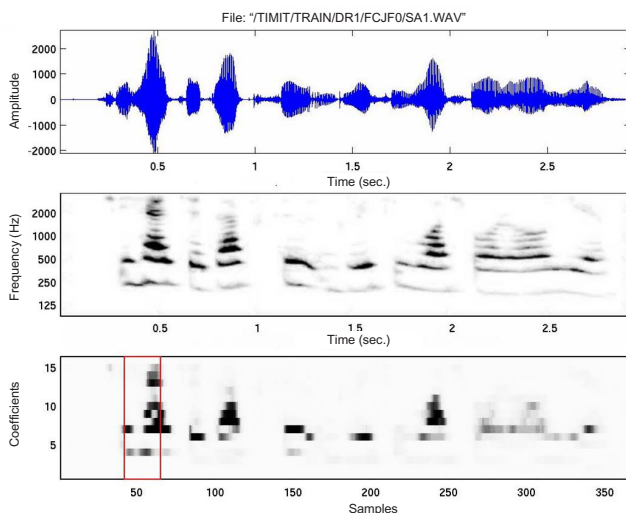
According to the previous considerations an experiment of phoneme classification was designed to evaluate the performance of a system that uses a cortical representation for this task. The speech data, corresponding to region DR1 of

**Table 1.** Distribution of patterns per class for training and test data

PHONEME	TRAIN		TEST	
	#	(%)	#	(%)
/b/	211	(3.26)	66	(3.43)
/d/	417	(6.45)	108	(5.62)
/jh/	489	(7.56)	116	(6.04)
/eh/	2753	(42.58)	799	(41.63)
/ih/	2594	(40.13)	830	(43.25)
Total	6464	(100.00)	1919	(100.00)

TIMIT corpus [19] for the set of five highly confusing phonemes /b/, /d/, /jh/, /eh/, /ih/, were used (See Table 1).

For each one of the emissions, sampled at 16 KHz, the corresponding auditory spectrogram was calculated from an early auditory model [20]. Then, the frequency resolution of the data was reduced so as to diminish its dimensions. After that, auditory spectrograms with a total of 64 frequency coefficients per time unit were obtained. Finally, by means of a sliding window of 32 ms in length at intervals of 8 ms, the set of spectro-temporal patterns that served as a base for the estimation of the dictionaries were obtained. In Figure 2, the main steps in this process, as well as the corresponding signals are shown.



**Fig. 2.** Main steps in the process used to generate the spectro-temporal patterns that serve as a basis for obtaining the STRF: sonogram (top), original auditory spectrogram (center) and low-resolution spectrogram (bottom). In this last representation, a section corresponding to the sliding window, from which each spectro-temporal pattern is generated, has been marked.

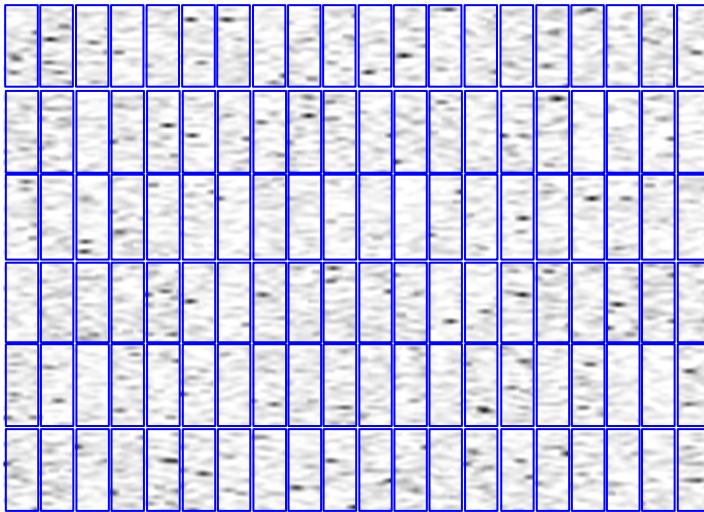
From these spectro-temporal patterns, different dictionaries of two-dimensional atoms were trained using (9) [21]. Several tests for both the complete and overcomplete cases of each configuration were conducted.

Once the STRF were obtained, the activation coefficients were calculated in an iterative form using (6) from the auditory spectrograms. For comparison purposes the *mel frequency cepstral coefficients* (MFCC) with an energy coefficient (MFCC+E) were calculated in the usual way for two consecutive frames, resulting in patterns in  $\mathbb{R}^{28}$  [22].

The classification was carried out for each experiment using an artificial neural network, namely a *multi-layer perceptron* (MLP). The network architecture consisted of one input layer, where the number of input units depended on the dimension of the patterns, one hidden layer, and one output layer of 5 units. The number of units in the hidden layer was varied, depending on the experiment.

## 4 Results and Discussion

An example of some of the STRFs obtained is shown in Figure 3. This case corresponds to the complete dictionary  $\Phi \in \mathbb{R}^{256 \times 256}$ , using patterns of 64x4. Several typical behaviors can be observed, which are useful for discriminating between the different phonemes used. The relative position of each element in the dictionary is related to its similarity with the other elements in the dictionary (in terms of the  $\ell_2$  norm of their differences). It is possible to observe that some STRF seem to act like detectors of diverse significant phonetic characteristics,



**Fig. 3.** Some spectro-temporal receptive fields as obtained from the patterns of 64x4 points of the early auditory representations. Each STRF has 4 KHz height and 32 ms width. The speech utterances were taken from five phonemes of TIMIT corpus, region DR1.

**Table 2.** Recognition rates of phoneme classification experiments with MLP using the representations generated by means of the early auditory models, the cortical representation obtained from the activation of the STRF, and MFCCs (best results in bold)

N <sup>o</sup>	EXPERIMENT	NETWORK	TRN	TST	/b/	/d/	/jh/	/eh/	/ih/	
1	Auditory	64x4	256/4/5	45.84	44.76	0.00	0.00	6.90	100.00	6.27
2			256/8/5	44.35	43.25	0.00	0.00	4.31	100.00	3.13
3			256/16/5	64.28	65.03	0.00	0.00	9.48	94.99	57.59
4			256/32/5	68.92	69.67	0.00	0.00	100.00	95.87	54.82
5			256/64/5	70.70	72.69	0.00	0.00	83.62	72.34	86.75
6			256/128/5	70.50	72.17	4.55	0.00	62.93	84.73	76.14
7			256/256/5	72.15	73.74	0.00	0.00	97.41	85.23	74.82
8			256/512/5	69.21	71.76	0.00	0.00	100.00	94.49	60.96
9	Cortical	64x4	256/4/5	77.04	75.72	40.91	56.48	97.41	84.86	69.16
10			256/8/5	79.64	<b>77.64</b>	46.97	62.96	93.97	84.86	72.77
11			256/16/5	75.60	76.08	65.15	51.85	97.41	89.99	63.73
12			256/32/5	79.72	74.73	65.15	67.59	98.28	79.22	68.80
13			256/64/5	87.27	76.86	74.24	66.67	95.69	88.24	64.82
14			256/128/5	100.00	<b>78.37</b>	72.73	70.37	96.55	78.35	77.35
15			256/256/5	98.10	77.07	65.15	71.30	91.38	87.11	67.11
16			256/512/5	99.92	<b>79.16</b>	71.21	69.44	92.24	80.35	78.07
17	Cortical	64x4x2	512/4/5	78.65	73.79	48.48	59.26	86.21	85.61	64.58
18			512/8/5	80.62	75.51	63.64	59.26	98.28	85.36	65.90
19			512/16/5	78.65	74.26	54.55	53.70	99.14	82.98	66.63
20			512/32/5	82.58	75.66	62.12	66.67	95.69	85.11	66.02
21			512/64/5	87.27	75.87	54.55	65.74	98.28	83.48	68.43
22			512/128/5	84.72	75.98	65.15	56.48	95.69	84.23	68.67
23			512/256/5	81.37	76.55	65.15	62.96	95.69	86.86	66.63
24			512/512/5	82.64	76.32	65.15	61.11	97.41	77.97	74.70
25	MFCC+E	14+14	28/28/5	77.39	77.28	46.51	75.38	91.11	80.56	74.40

e.g. unique frequencies, stable speech formant patterns, changes in the speech formants, unvoiced or fricative components, and well-located patterns in time and/or frequency.

The results of the experiments described in the previous section are detailed in Table 2. As can be seen from this table, the results of classification on the training and test data for the cortical representation are better than those obtained when using the direct (or early) auditory representation. For the latter representation, some of the classification results are apparently globally good, however, when the individual phoneme classification rates (exhibited in the right-most columns of the table) are examined, only two or three phonemes are in fact correctly classified (see experiments N<sup>o</sup> 1-8). This problem arises because of a local minimum error solution that the cortical representation avoids (see the uneven pattern distribution in Table 1).

Moreover, the results for the cortical representation are better than those obtained using the classical MFCC representation on this task (see experiments N° 16 and 25 in Table 2). Another important aspect is that the performance is satisfactory for relatively small network architectures in relation to the pattern dimensions. This aspect corroborates the hypothesis that the classes are better separated in this new higher dimensional space, and therefore a simpler classifier can complete the task successfully.

The statistical significance of these results was evaluated considering the probability that the classification error of a given classifier  $\epsilon$  is smaller than the one of the reference system  $\epsilon_{ref}$ . In order to make this estimation, the statistical independence of the errors for each frame was assumed, and the binomial distribution of the errors was modeled by means of a Gaussian distribution (this is possible because a sufficiently large number of test frames is given). Therefore, comparing the MFCC with the best result of the cortical representation, the  $Pr(\epsilon_{ref} > \epsilon) > 92\%$  was obtained.

Harpur [13] conducted some simple experiments of phoneme classification using low entropy codes, with only positive coefficients generated from a filter bank. However, experiments using more complex models of the auditory receptive fields – such as the ones that appear here – have not been previously reported.

## 5 Conclusions

In this work, a new approach to speech feature extraction, based on a biological metaphor, has been proposed and applied to a phoneme classification task. This approach first finds an early auditory representation of the speech signal at the auditory nerve level. Then, based on analogies established with neuro-sensorial systems, an optimal dictionary is estimated from the auditory spectrograms of speech data.

The method finds a set of atoms which can be related to the spectro-temporal receptive fields of the auditory cortex, and which prove capable of functioning like detectors of important phonetic characteristics. It is worthwhile mentioning, for example, the detection of events based on highly localized spectro-temporal features, such as relatively stationary segments, different types of formant evolution, and non-harmonic zones.

Using representations provided by the method as the input patterns, multi-layer perceptrons were trained as phoneme classifiers. The results obtained improve those of both early auditory and standard MFCC representations. The objective was not to find the best possible classifier, but rather to demonstrate the feasibility of the proposed method. Obviously, further experimentation is called for.

Another interesting issue, that also remains to be explored in future works, is the evaluation of the robustness of this type of representation in the presence of additive noise.



## Acknowledgements

The authors wish to thank: the *Universidad Nacional de Litoral* (with UNL-CAID 012-72), the *Agencia Nacional de Promoción Científica y Tecnológica* (with ANPCyT-PICT 12700 & 25984) and the *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET) from Argentina and the *Consejo Nacional de Ciencia y Tecnología* (CONACYT) and the *Secretaría de Educación Pública* (SEP) from Mexico, for their support.

## References

1. Greenberg, S.: The ears have it: The auditory basis of speech perception. In: Proceedings of the International Congress of Phonetic Sciences, vol. 3, pp. 34–41 (1995)
2. Rufiner, H., Goddard, J., Rocha, L.F., Torres, M.E.: Statistical method for sparse coding of speech including a linear predictive model. *Physica A: Statistical Mechanics and its Applications* 367(1), 231–250 (2006)
3. Delgutte, B.: Physiological models for basic auditory percepts. In: Hawkins, H., McMullen, T., Popper, A., Fay, R. (eds.) *Auditory Computation*, Springer, New York (1996)
4. Simon, J.Z., Depireux, D.A., Klein, D.J., Fritz, J.B., Shamma, S.A.: Temporal symmetry in primary auditory cortex: Implications for cortical connectivity. *Neural Computation* 19(3), 583–638 (2007)
5. Kording, K.P., Konig, P., Klein, D.J.: Learning of sparse auditory receptive fields. In: Proc. of the International Joint Conference on Neural Networks (IJCNN 2002), Honolulu, HI, United States, vol. 2, pp. 1103–1108 (2002)
6. Klein, D., Konig, P., Kording, K.: Sparse Spectrotemporal Coding of Sounds. *EURASIP Journal on Applied Signal Processing* 2003(7), 659–667 (2003)
7. Oja, E., Hyvarinen, A.: *Independent Component Analysis: A Tutorial*. Helsinki University of Technology, Helsinki (2004)
8. Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via  $l_1$  minimization. *Proceedings of the National Academy of Sciences* 100(5), 2197–2202 (2003)
9. Theunissen, F., Sen, K., Doupe, A.: Spectro-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds. *J. Neuroscience* 20, 2315–2331 (2000)
10. deCharms, R., Blake, D., Merzenich, M.: Optimizing sound features for cortical neurons. *Science* 280, 1439–1443 (1998)
11. Olshausen, B.: Sparse codes and spikes. In: Rao, R.P.N., Olshausen, B.A., Lewicki, M.S. (eds.) *Probabilistic Models of the Brain: Perception and Neural Function*, pp. 257–272. MIT Press, Cambridge (2002)
12. Olshausen, B., Field, D.: Emergence of simple cell receptive field properties by learning a sparse code for natural images. *Nature* 381, 607–609 (1996)
13. Harpur, G.F.: *Low Entropy Coding with Unsupervised Neural Networks*. PhD thesis, Department of Engineering, University of Cambridge, Queens' College (1997)
14. Hyvärinen, A.: *Sparse code shrinkage: Denoising of nongaussian data by maximum-likelihood estimation*. Technical report, Helsinki University of Technology (1998)
15. Lewicki, M., Olshausen, B.: A probabilistic framework for the adaptation and comparison of image codes. *Journal of the Optical Society of America* 16(7), 1587–1601 (1999)

16. Abdallah, S.A.: Towards music perception by redundancy reduction and unsupervised learning in probabilistic models. PhD thesis, Department of Electronic Engineering, King's College London (2002)
17. Barlow, H.: Redundancy reduction revisited. *Network: Computation in Neural Systems* (12), 241–253 (2001)
18. Kwon, O.W., Lee, T.W.: Phoneme recognition using ICA-based feature extraction and transformation. *Signal Processing* 84(6), 1005–1019 (2004)
19. Garofolo, Lamel, Fisher, Fiscus, Pallett, Dahlgren.: DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus Documentation. Technical report, National Institute of Standards and Technology (1993)
20. Yang, X., Wang, K., Shamma, S.A.: Auditory representations of acoustic signals. *IEEE Trans. Inf. Theory. Special Issue on Wavelet Transforms and Multiresolution Signal Analysis* 38, 824–839 (1992)
21. Lewicki, M., Sejnowski, T.: Learning overcomplete representations. In: *Advances in Neural Information Processing 10 (Proc. NIPS 1997)*, pp. 556–562. MIT Press, Cambridge (1998)
22. Deller, J., Proakis, J., Hansen, J.: *Discrete Time Processing of Speech Signals*. Macmillan Publishing, New York (1993)

# Using Adaptive Filter and Wavelets to Increase Automatic Speech Recognition Rate in Noisy Environment

José Luis Oropeza Rodríguez and Sergio Suárez Guerra

Center for Computing Research, National Polytechnic Institute,  
Juan de Dios Batiz esq Miguel Othon de Mendizabal s/n, P.O. 07038, Mexico  
joropeza@cic.ipn.mx, ssuarez@cic.ipn.mx

**Abstract.** This paper shows results obtained in the Automatic Speech Recognition (ASR) task for a corpus of digits speech files with a determinate noise level immerse. In the experiments, we used several speech files that contained Gaussian noise. We used HTK (Hidden Markov Model Toolkit) software of Cambridge University in the experiments. The noise level added to the speech signals was varying from fifteen to forty dB increased by a step of 5 units. We used an adaptive filtering to reduce the level noise (it was based in the Least Measure Square –LMS- algorithm) and two different wavelets (Haar and Daubechies). With LMS we obtained an error rate lower than if it was not present and it was better than wavelets employed for this experiment of Automatic Speech Recognition. For decreasing the error rate we trained with 50% of contaminated and originals signals to the ASR system. The results showed in this paper are focused to try analyses the ASR performance in a noisy environment and to demonstrate that if we are controlling the noise level and if we know the application where it is going to work, then we can obtain a better response in the ASR tasks. Is very interesting to count with these results because speech signal that we can find in a real experiment (extracted from an environment work, i.e.), could be treated with these technique and we can decrease the error rate obtained. Finally, we report a recognition rate of 99%, 97.5% 96%, 90.5%, 81% and 78.5% obtained from 15, 20, 25, 30, 35 and 40 noise levels, respectively when the corpus mentioned before was employed and LMS algorithm was used. Haar wavelet level 1 reached up the most important results as an alternative to LMS algorithm, but only when the noise level was 40 dB and using original corpus.

**Keywords:** Automatic Speech Recognition, Haar wavelets, Daubechies wavelet, Least Measure Square and noisy speech signal, noisy reduction.

## 1 Introduction

Speech and language are perhaps the most evident expression of human thought and intelligence the creation of machines that fully emulate this ability poses challenge that reach far beyond the present state of the art.

The speech recognition field has been fruitfully and productively benefited from sciences as diverse as computer science, electrical engineering, biology, psychology, linguistics, statistics, philosophy, physics and mathematics among others. The interplay between different intellectual concerns, scientific approaches, and models, and its potential impact in society make speech recognition one of the most challenging, stimulating, and exciting fields today.

Exhaustive search in very large vocabularies is typically unmanageable. Instead, one must turn to smaller sub-word units (phonemes, syllables, triphonemes, etc.), which may be more ambiguous and harder to detect and recognize.

The different sources of variability that can affect speech determine most of difficulties of speech recognition. During speech production the movements of different articulators overlap in time for consecutive phonetic segments and interact with each other. As a consequence, the vocal tract configuration at any time is influenced by more than one phonetic segment. This phenomenon is known as coarticulation. The principal effect of the coarticulation is that the same phoneme can have very different acoustic characteristics depending on the context in which it is uttered [Farnetani 97].

Speech recognition-system performance is also significantly affected by the acoustic confusability or ambiguity of the vocabulary to be recognized. A confusable vocabulary requires detailed high performance acoustic pattern analysis. Another source of recognition-system performance degradation can be described as variability and noise.

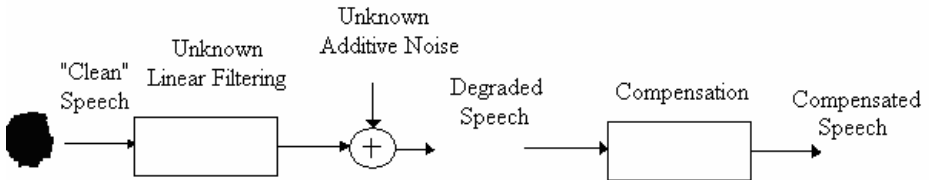
State-of-the-art ASR systems work pretty well if the training and usage conditions are similar and reasonably benign. However, under the influence of noise, these systems begin to degrade and their accuracies may become unacceptably low in severe environments [Deng and Huang 2004]. To remedy this noise robustness issue in ASR due to the static nature of the HMM parameters once trained, various adaptive techniques have been proposed. A common theme of these techniques is the utilization of some form of compensation to account for the effects of noise on the speech characteristics. In general, a compensation technique can be applied in the signal, feature or model space to reduce mismatch between training and usage conditions [Huang et al. 2001].

## 2 Characteristics and Generalities

Speech recognition systems work reasonably well in quiet conditions but work poorly under noisy conditions or distorted channels. For example, the accuracy of a speech recognition system may be acceptable if you call from the phone in your quiet office, yet its performance can be unacceptable if you try to use your cellular phone in a shopping mall. The researchers in the speech group are working on algorithms to improve the robustness of speech recognition system to high noise levels channel conditions not present in the training data used to build the recognizer

Robustness in speech recognition refers to the need to maintain good recognition accuracy even when the quality of the input speech is degraded, or when the acoustical, articulatory, or phonetic characteristics of speech in the training and testing environments differ. Obstacles to robust recognition include acoustical degradations produced by additive noise, the effects of linear filtering, nonlinearities in transduction or

transmission, as well as impulsive interfering sources, and diminished accuracy caused by changes in articulation produced by the presence of high-intensity noise sources. Some of these sources of variability are illustrated in Figure 1. Speaker-to-speaker differences impose a different type of variability, producing variations in speech rate, co-articulation, context, and dialect, even systems that are designed to be speaker independent exhibit dramatic degradations in recognition accuracy when training and testing conditions differ [Cole & Hirschman 92].



**Fig. 1.** Schematic representation of some of the sources of variability that can degrade speech recognition accuracy, along with compensation procedures that improve environmental robustness

As speech recognition and spoken language technologies are being transferred to real applications, the need for greater robustness in recognition technology is becoming increasingly apparent. Nevertheless, the performance of even the best state-of-the-art systems tends to deteriorate when speech is transmitted over telephone lines, when the signal-to-noise ratio (SNR) is extremely low (particularly when the unwanted noise consists of speech from other talkers), and when the speaker's native language is not the one with which the system was trained.

### 3 Automatic Speech Recognition Systems

Automatic Speech recognition systems generally assume that the speech signal is a realization of some message encoded as a sequence of one or more symbols. The ASR is constitutive by: training and recognition stages.

Voice is a static procedure that can to have a duration time between 80-200 ms. a simple but effective mathematical model of the physiological voice production process is the excitation and vocal tract model.

The excitation signal is assumed periodic with a period equal to the pitch for vowels and other voiced sounds, while for unvoiced consonants, the excitation is assumed white noise, i.e. a random signal without dominant frequencies. The excitation signal is subject to spectral modifications while it passes through the vocal tract that has an acoustic effect equivalent to linear time invariant filtering. The model is relevant because, for each type of excitation, a phoneme (or another structural linguistic) is identified mainly by considering the shape of the vocal tract. Therefore, the vocal tract configuration can be estimated by identifying the filtering performed by the tract vocal on the excitation. Introducing the power spectrum of the signal  $P_x(\omega)$ , of the excitation  $P_v(\omega)$  and the spectrum of the vocal tract filter  $P_h(\omega)$ , we have:

$$P_x(\omega) = P_v(\omega)P_h(\omega) \tag{1}$$

The speech signal (continuous, discontinuous or isolated) is first converted to a sequence of equally spaced discrete parameter vectors. This sequence of parameter vectors is assumed to form an exact representation of the speech waveform on the basis that for the duration covered by a single vector (typically 10-25 ms) the speech waveform can be regarded as being stationary. Although it is not strictly true, it is a reasonable approximation. Typical parametric representations in common use are smoothed spectra or linear predictive coefficients plus various other representations derived from these. The database employed consists of ten digits (0-9) for the Spanish language. Many of the operations performed by HTK (Hidden Markov Model Toolkit) which involve speech data assumes that the speech is divided into segments and each segment has a name or label. The set of labels associated with the speech data will be the same as corresponding speech file but a different extension.

### 4 Hidden Markov Models

As we know, HMMs mathematical tool applied for speech recognition presents three basic problems [Rabiner and Biing-Hwang, 1993] y [Zhang 1999]. For each state, the HMMs can use since one or more Gaussian mixtures both to reach high recognition rate and modeling vocal tract configuration in the Automatic Speech Recognition.

#### Gaussian mixtures

Gaussian Mixture Models are a type of density model which comprise a number of functions, usually Gaussian. These component functions are combined to provide a multimodal density. They can be employed to model the colors of an object in order to perform tasks such as real-time color-based tracking and segmentation. In speech recognition, the Gaussian mixture is of the form [Bilmes 98] [Resch, 2001a], [Resch, 2001b], [Kamakshi et al., 2002] and [Mermelstein, 1975].

:

$$g(\mu, \Sigma)(x) = \frac{1}{\sqrt{2\pi^d} \sqrt{\det(\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{2}$$

Equation 12 shows a set of Gaussian mixtures:

$$gm(x) = \sum_{k=1}^K w_k * g(\mu_k, \Sigma_k)(x) \tag{3}$$

In 12, the summarize of the weights give us

$$\sum_{i=1}^K w_i = 1 \quad \forall \quad i \in \{1, \dots, K\} : w_i \geq 0 \tag{4}$$

#### Viterbi Training

We used Viterbi training, in this a set of training observations  $O^r$ ,  $1 \leq r \leq R$  is used to estimate the parameters of a single HMM by iteratively computing Viterbi

alignments. When used to initialise a new HMM, the Viterbi segmentation is replaced by a uniform segmentation (i. e. each training observation is divided into N equal segments) for the first iteration.

## Wavelets Transform

Classically, Fourier and related representations have been used widely in image processing applications. The noise removal has been done using the Wiener filter, which is derived by assuming a signal model of uncorrelated Gaussian-distributed coefficients in the Fourier domain and utilizes second-order statistics of the Fourier coefficients.

The statistics of many natural images are simplified when they are decomposed via wavelet transform. Recently, many researchers have found that statistics of order greater than two can be utilized in choosing a basis for images. [Field 94] has shown that the coefficients of frequency subbands of natural scenes have much higher kurtosis than a Gaussian distribution. Based on this observation [Simoncelli 96] suggested an algorithm for noise removal using a non-Gaussian marginal model. [Simoncelli 99] also developed a joint non-Gaussian Markov model utilizing the dependence between the coefficients of subbands.

The Haar wavelet is the first known wavelet and was proposed in 1990 by Alfred Harr. Note that the term wavelet was coined much later. As a special case of the Daubechies wavelet, it is also known as D2.

The Haar wavelet is also the simplest possible wavelet. The disadvantage of the Haar wavelet is that it is not continuous and therefore not differentiable. The Haar wavelet can also be described as a step function  $f(x)$  with

$$f(x) = \begin{cases} 1 & \text{para } 0 \leq x < 1/2 \\ -1 & \text{para } 1/2 \leq x < 1 \\ 0 & \text{para en otro caso} \end{cases} \quad (5)$$

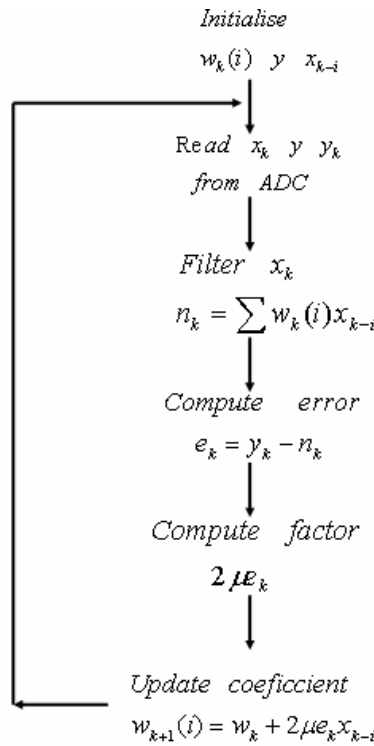
Daubechies wavelets are a family of orthogonal wavelets defining a discrete wavelet transform and characterized by a maximal number of vanishing moments for some given support. With each wavelet type of this class, there is a scaling function (also called mother wavelet) which generates an orthogonal multiresolution analysis.

The selected subband was the 3 in all case, because the high level recognition was better for this case.

## 5 Experiments and Results

The evaluation of the algorithm proposed involved clustering a set of speech data consisting of 100 isolated patterns from a digits vocabulary. The training patterns (and a subsequent set of another 200 independent testing pattern) were recorded in a room free of noise. Only one speaker provided the training and testing data. All training and test recordings were made under identical conditions. The 200 independent testing

patterns was addition with a level noise, we obtained a total of 1200 new sentences contaminated (200 per noise level, that is because we used 6 noise levels). After that, we used an adaptive filter to reduce that noise level and the results are shown below, then we obtained another 1200 sentences. Finally, we made experiments with a total of 2600 sentences (between noisy, filtered and clean sentences) of speech signal. Figure 2 shows the adaptive filter algorithm employed. For each corpus created, we used three databases test to recognition task: with same characteristics, noisy and filtered. All sentences were recorded at 16 kHz frequency rate, 16 bits and mono-channel. We use MFCCs (Mel Frequency Cepstral Coefficients) with 39 characteristics vectors (differential and energy components). A Hidden Markov Model with 5 states and 1 Gaussian Mixture per state.



**Fig. 2.** Adaptive filter algorithm

This algorithm stop when the error is lest than 0.9%.

Table 1 shows the results obtained when we used a noisy corpus to training the ASR. A total of 600 speech sentences were analyzed.

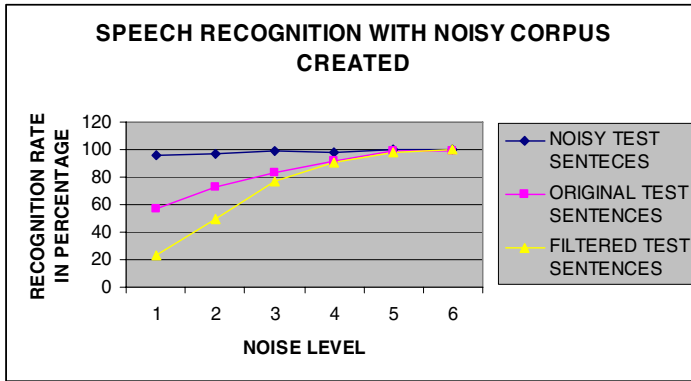
As we can see, when we used a noisy corpus like we hoped, recognition level with noisy database was adequately. When we used high S/N rate (25, 30, 35 and 40 dB),



the recognition rate was increased. It is important because it significance that the noisy corpus is a good reference. Figure 3 shows a histogram related with the table contents.

**Table 1.** Results obtained with noisy corpus created

	speech recognition with noisy corpus created					
	noise level					
Speech signal recognized	15	20	25	30	35	40
Noisy	95,5	96,5	98,5	98	99,5	99,5
Original	57	72,5	83,5	91,5	99	99
Filtered	23	50	76,5	90,5	98	99,5



**Fig. 3.** Graphic representation using noisy corpus created

Table 2 shows the results obtained when we used a noisy and clean corpus to training the ASR. A total of 600 (300 noisy and 300 clean) speech sentences were analyzed.

**Table 2.** Results obtained with noisy and clean corpus created

	speech recognition with noisy and clean corpus created					
	noise level					
Speech signal recognized	15	20	25	30	35	40
Noisy	98,5	98	99,5	99	99,5	99,5
Original	19	34	84	91,5	96,5	99
Filtered	78,5	81	90,5	96	95,7	99

As we can see, when we used a corpus compound by noisy and original signals, the recognition rate for filtered speech signal was increased considerably. Figure 2 shows that.

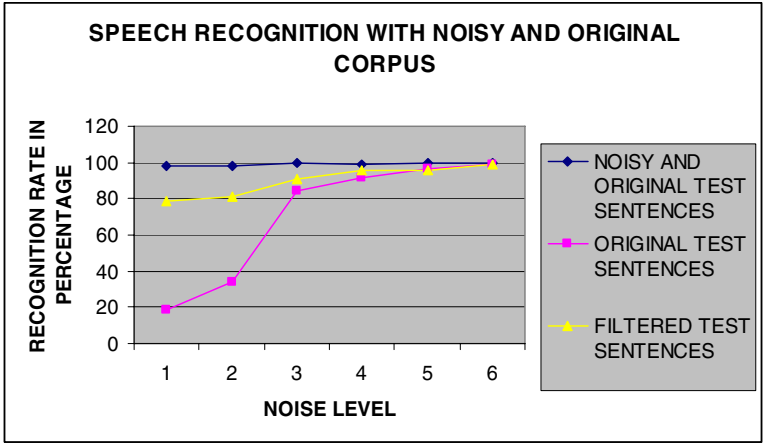


Fig. 4. Graphic representation using noisy and original corpus

Table 3 shows the results obtained when we used a clean corpus to training the ASR. A total of 600 speech sentences were analyzed.

With the original corpus the results was not satisfactory, although the recognition rate with filtered signals was better than noisy signals, it was poor and not enough to be considered important as figure 5 shows.

Table 3. Results obtained with clean corpus created

Speech signal recognized	speech recognition with clean corpus created					
	noise level					
	15	20	25	30	35	40
Noisy	99,5	99,5	99,5	99,5	99,5	99,5
Original	16	21,5	18	43	70,5	87
Filtered	18,5	29	33,5	56	99,5	86,5

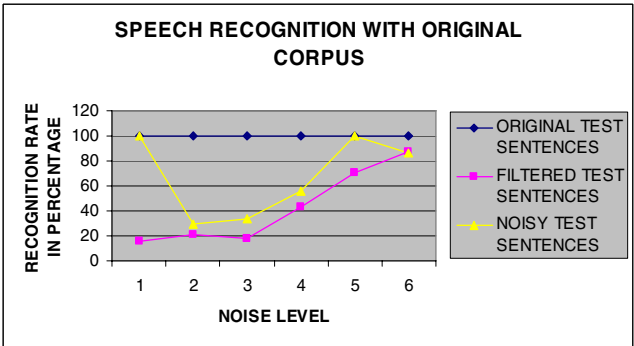


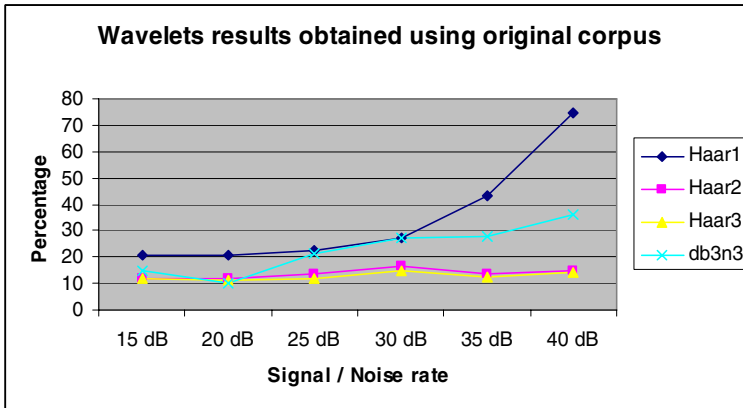
Fig. 5. Graphic representation using original corpus created

Finally, we probed different wavelets to try to determine better results than we obtained above. The results were not that we hoped.

**Table 4.** Results obtained with clean corpus created and wavelets

Atenuación	Haar1	Haar2	Haar3	db3n3
15 dB	20,5	12	12	15
20 dB	21	12	11,5	10
25 dB	22,5	13,5	12	21,5
30 dB	27,5	16,5	15	27,5
35 dB	43,5	13,5	12,5	28
40 dB	74,5	15	14	36

As we can see in figure 6, only Haar 1 wavelet at 40 dB had a high performance in ASR rate. We consider that results obtained were failed because noisy level selected before to apply wavelet transform must be changed. But we consider that it only can not help us so much.



**Fig. 6.** Graphic representation for ASR using wavelets and original corpus

## 6 Conclusions and Future Works

The results shown in this paper demonstrate that we can use an adaptive filter to reduce the noise level in an automatic speech recognition system (ASRS) for the Spanish language. The use of this paradigm is not new but with this experiment we propose to reduce the problems find out when we tread with real speech signals. MFCCs and CDHMMs (Continuous Density Hidden Markov Models) were used for training and recognition, respectively. First, when we used database test with the same characteristics that corpus training a high performance was reached out, but when we used the clean speech database our recognition rate was poor. The most important results extracted of this experiment were when the clean speech was fixed with noisy speech, when we used filtered speech we obtained a high performance in our ASR.

For that, our conclusion is that if we want to construct an ASR immerse in a noisy environment, it is going to have a high performance if we included in our database training clean and noisy speech signal. So, if we known the Signal/Noise ratio and it's greater than 35%, we can use the filtered signal in an ASR without problems. For future works is recommendable try to probe the results obtained using another methods employed to reduce noise into signal (wavelets i. e.), and extract the results.

## References

- [Bilmes 98] Bilmes, J.A.: A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. International Computer Science Institute, Berkeley, CA (1998)
- [Cole & Hirschman 92] Cole, R.A., Hirschman, L., et al.: Workshop on spoken language understanding. Technical Report CSE 92-014, Oregon Graduate Institute of Science & Technology, Portland, OR, USA (September 1992)
- [Deng, Li. and Huang, X. (2004)] Challenges in Adopting Speech Recognition. *Communications of the ACM* 47(1), 69–75 (2004)
- [Farnetani 97] Farnetani, E.: Coarticulation and connected speech processes. In: Hardcastle, W., Laver, J. (eds.) *The Handbook of Phonetic Sciences*, pp. 371–404. Blackwell, Oxford (1997)
- [Field 94] Field, D.J.: What is the goal of sensory coding. *Neural Computation* 6, 559–601 (1994)
- [Huang & Lee,93] Huang, X.D., Lee, K.F.: On speaker-independent, speaker-dependent, and speaker-adaptive speech recognition. *IEEE Transactions on Speech and Audio Processing* 1(2), 150–157 (1993)
- [Huang, C., Wang, H. and Lee, C. (2001)] An ASR Incremental Stochastic Matching Algorithm for Noisy Speech Recognition. *IEEE Trans. Speech and Audio Processing* 9(8), 866–873
- [Kamakshi et al. 2002] Prasad, K.V., Nagarajan, T., Murthy Hema, A.: Continuous Speech Recognition Using Automatically Segmented Data at Syllabic Units. Department of Computer Science and Engineering. Indian Institute of Technology. Madras, Chennai 600-636 (2002)
- [Mermelstein 1975] Mermelstein, P.: Automatic Segmentation of Speech into Syllabic Units. *Haskins Laboratories, New Haven, Connecticut* 06510 58(4), 880–883 (1975)
- [Rabiner and Biing-Hwang 1993] Rabiner, L., Juang, B.-H.: *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs (1993)
- [Simoncelli 96] Simoncelli, E.P., Anderson, E.H.: Noise removal via Bayesian wavelet coring. In: *Proceedings of the 3rd IEEE International Conference on Image Processing*, vol. 1, pp. 379–382 (1996)
- [Simoncelli 99] Simoncelli, E.P.: Bayesian Denoising of Visual Image in the Wavelet Domain. In: Muller, P., Vidakovic, B. (eds.) *Bayesian Inference in Wavelet Based Models*, ch. 18. pp. 291–308. Springer, Heidelberg (1999)
- [Zhang 1999] Zhang, J.: "On the syllable structures of Chinese relating to speech recognition", Institute of Acoustics, Academia Sinica Beijing, China (1999)

# Spoken Commands in a Smart Home: An Iterative Approach to the Sphinx Algorithm

Michael Denkowski, Charles Hannon, and Antonio Sanchez

Texas Christian University,  
Fort Worth, Texas, 76129, USA

{m.j.denkowski, c.hannon, a.sanchez-aguilar}@tcu.edu

**Abstract.** An algorithm for decoding commands spoken in an intelligent environment through iterative vocabulary reduction is presented. Current research in the field of speech recognition focuses primarily on the optimization of algorithms for single pass decoding using large vocabularies. While this is ideal for processing conversational speech, alternative methods should be explored for different domains of speech, specifically commands issued verbally in an intelligent environment. Such commands have both an explicitly defined structure and a vocabulary limited to valid task descriptions. We propose that a multiple pass context-driven decoding scheme utilizing dictionary pruning yields improved accuracy; this occurs when one deals with command structure and a reduced vocabulary. Each iteration incorporates the hypothesis of the previous into its decoding scheme by removing unlikely words from the current language model. We have applied this decoding method to a comprehensive set of spoken commands through the use of Sphinx-4, an Automatic Speech Recognition (ASR) engine using the Hidden Markov Model (HMM). When decoding via HMM, multiple previous states are used to determine the current state, thus utilizing context to aid in intelligent recognition. Our results show that within a fixed domain, multiple pass decoding yields recognition accuracy. Further research must be conducted to optimize practical context driven decoding and to apply the method to larger domains, primarily those of intelligent environments.

*“Successful software always gets changed”.*

Frederick P. Brooks, Jr.

*“Optimism is an occupational hazard of programming:  
testing is the treatment”*

Kent Beck

## 1 Introduction

Current research in the field of speech recognition focuses primarily on the optimization of algorithms for single pass decoding using large vocabularies [1]. Yet any intelligent environment where commands are spoken, there naturally exists a domain restriction on both syntax and vocabulary. Current speech recognition technology does not take full advantage of this restriction. In a smart home environment, communication between the users and the home controller provides a control niche where such restrictions are valid [2]. In such a case, recognition of the spoken language is a necessary condition regardless the speaker. Yet training is the

means upon which most commercial systems rely heavily as a mechanism for raising accuracy, thus limiting the recognition of casual users[3]. Therefore such methods do not completely fulfill the needs of intelligent environments such as homes with visitors.

In this paper we discuss the implementation of a speech recognition system for a smart home controller, and therefore concentrate on providing robustness to an available speech system within the restriction of limited set of commands to control a smart home. It is important to note that the system must be accessible to multiple user speech input with highly reliable results. In our research we use the successful Sphinx-4 system in its open source Java version [4] designed and implemented specifically for research development. The proposed context driven multiple pass approach is tested and the results of our research are reported; at the end we present some suggestions for further development.

## 2 Related Research

### 2.1 Spoken Language Research and the Sphinx

Our research began with the use of the successful CMU Sphinx system. Specifically, we use the open source Java implementation of Sphinx-4; the system was jointly developed by Carnegie Mellon University, Sun Microsystems Laboratories, and Mitsubishi Electric Research Laboratories. It is a highly modular system that supports the well known “Hidden Markov” (HMM) acoustic models with all standard types of language models and multiple search strategies. Figure 1 shows the overall architecture of the Sphinx-4 decoder. The front end module is in charge of creating a parameter vector of the speech signals, which communicates the derived features to the decoding block. Such a module has three components: the search manager, the linguist, and the acoustic scorer. The modules work in in tandem to perform the over all decoding. Figure 2 shows a detailed representation of the front-end module consisting of several communicating blocks, each with an input and an output. All blocks are input/output linked to the previous and successor modules. Whenever a block is ready for more data, it reads data from the predecessor, and interprets it to find out if the incoming information is speech data or a control signal. The control signal might indicate beginning or end of speech – important for the decoder – or might indicate data dropped or some other problem. It is important to note that when the data is speech, it is processed and the output is buffered, waiting for the successor block to request it.

The decoder block consists of the following three modules: search manager, linguist, and acoustic scorer. The function of the search manager is to create and search a tree of possibilities for the best hypothesis for decoding. The construction of the search tree is done based on information obtained from the linguist tables and the acoustic scorer to obtain the best possible hypothesis. Processing is done using a set of token trees, a common scheme in speech recognition. The tree consists of a set of tokens that contain information about the search and provides a complete history of all active paths in the search. Tokens contain an overall acoustic and language score value of the path at a given point, following the HMM model of reference, thus

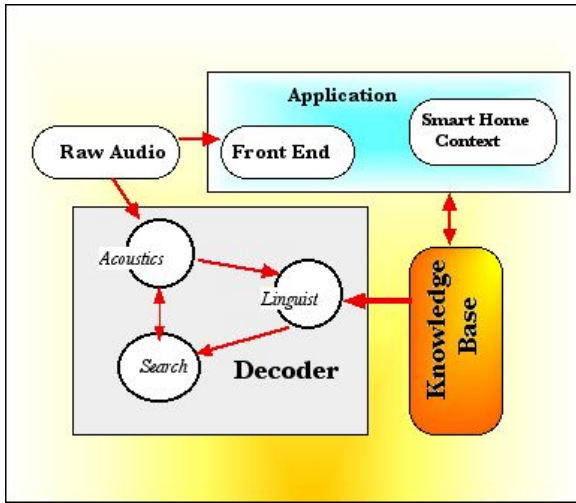


Fig. 1. Extensive communication links in Sphinx-4

allowing the manager to completely categorize a token as a context-dependent phonetic unit consisting of a pronunciation word and a grammar state. The reader is directed to reference [5] for a complete description of the Sphinx system, which is beyond the scope of this paper. Here let us simply state that by modifying the topology of the Sentence in the HMM, the footprint, perplexity, speed and recognition, accuracy can be affected.

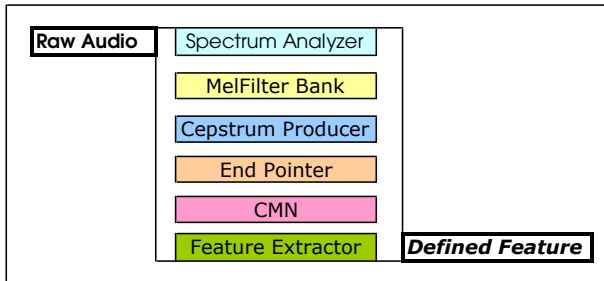


Fig. 2. Complex audio processing in Sphinx-4

## 2.2 Java Development

As stated before, our development and tests were performed using the Sphinx-4 speech recognizer, an open source project of Carnegie Mellon University. The system, written entirely in the Java programming language, is chosen for its high base success rate, flexibility, and modularity. The Sphinx-4 system and source code are freely obtainable from the Sphinx Project website. Due to its flexibility, it has even

been implemented for hand held devices as reported in [6]. The modularized design of Sphinx-4 allows compilation strategies to be modified without changing other aspects of the search; as stated by its developers, “Linguists can be broadly classified as either static or dynamic. A static linguist creates the entire SentenceHMM statically, prior to recognition, while the dynamic linguist only creates portions of the SentenceHMM that are relevant to the current set of hypotheses being evaluated. Notwithstanding the logistic constraints, the actual SentenceHMM structure itself is independent of whether the linguist is static or dynamic” [5]. Thus we were able to access the various jar files of previously developed software and integrate them with our new developments. A final point here is that although much has been said about Java as poor performance development tool, it has been reported [7] that for the case of speech systems this is not the case. Our additions to the existing system are also written in the Java programming language.

### 2.3 Smart Homes

In this day, Smart Homes and other intelligent spaces continue to entice us with their promise of anticipating and meeting our needs as they unobtrusively adapt to our changing preferences and goals. The delivery of this promise, however, has met with limited success in terms of functionality and consumer acceptance. Many commercial and academic efforts are in progress to create true smart home systems, and, to a lesser extent, to understand what customers really need. Therefore, our current work is focused on establishing a Smart Home Research lab based on integrating diverse developments in a cohesive manner and to meet real customer needs. We are interested in the high-level reasoning necessary to exploit the next generation of smart home devices. Specifically, we seek the "sweet spot" of automation between today's comprehensible systems of limited flexibility and potentially powerful (but unproven) autonomous systems that are error-prone and complicated. Finding this sweet spot involves investigating approaches for decision-making, adaptation, representing domain-specific knowledge, and new user interfaces. In any case the [8] the operation of the smart home must prove cohesive and simple in order to benefit its users.

There are many reasons to choose smart homes as the primary application for our lab. Among them; they represent an easily understandable domain, and have an appealing factor to induce young researchers to get involved. They also represent a useful application in society with a broad breath factor of the various AI areas of research.

## 3 Research Objectives and Implementation

### 3.1 Objectives and Working Hypothesis

With the background of spoken commands and software integration in smart home environment, the objective of our research is to validate the feasibility of context



driven speaker input recognizer. The working hypothesis of the paper is succinctly stated as follows:

***Hypothesis:*** *If the speech command decoder makes external multiple passes to prune out unlikely meaning based on previous building words, accuracy can be significantly improved thus obtaining higher confidence ratings, eventually determining the right command.*

The rationale of the hypothesis stems from the consideration that those words which are syntactically incompatible within a command can be eliminated from the vocabulary. The result will be correct, and thus more likely to be truly accurate. The implementation of this external context driven multiple pass algorithm is done by modifying the multi pass executed in Sphinx 4 as described before.

Before describing our modification, let us mention that in the standard Sphinx4 recognizer, the Search Manager, is responsible for determining the paths for the best hypothesis. Sphinx includes many search managers with varying degrees of complexity. However, as suggested by its developers, this is the part of the system which greatly welcomes third party improvements, as the accuracy and speed of recognition hinges on it. Therefore its internal multiple pass search manager allows Sphinx greater accuracy, but is still limited to the probability-based language models internal to system.

Concerning our context driven modification, we take the end result of Sphinx and prune possible words based on syntax, which is external to the Sphinx system, then correctly map syntax for commands which can be different from probability-based language models. Here we make multiple passes with the system based on this external syntax factor. For the time being the system as such is completely external to the standard Sphinx. In future generations of this research we might think on closely integrating this context based pruning to the internal multi pass Sphinx system.

### **Context Based Decoding**

Our algorithm is designed to iteratively prune the speech recognizer's vocabulary based on the hypotheses of previous passes.

- An initial pass is made, yielding a hypothesis
- The word with the highest confidence that can also be correctly located in a command is fixed
- The syntax map is then updated, removing paths to impossible words
- In turn, the syntax map updates the language model which the decoder references
- Another pass is made, and the process repeats until all words are set.

Figure 3 illustrates possible syntax paths after the second decoding pass of spoken command “radio volume down”. The first decoding pass fixed the word “volume”, the second fixing “radio”. As a result, only paths including words “radio” and “volume” remain possible.

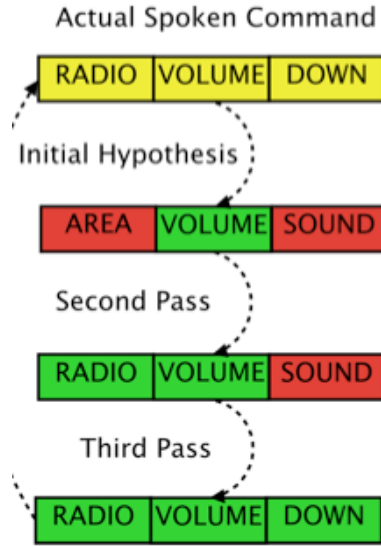


Fig. 3. Context driven multi pass algorithm

## 4 Results

Before describing the results, a word on the experiments executed is important. Each data set was recorded under the same conditions: the subjects spoke into a unidirectional microphone at a proximity comparable to that of a well placed lapel microphone used in a smart home demonstration. Each utterance lasted between 2 and 4 seconds, containing a single valid command. Each data set consisted of 15 commands assembled from the pool of valid devices, instructions, and modifiers in our smart home environment. The command set was constant for all test subjects.

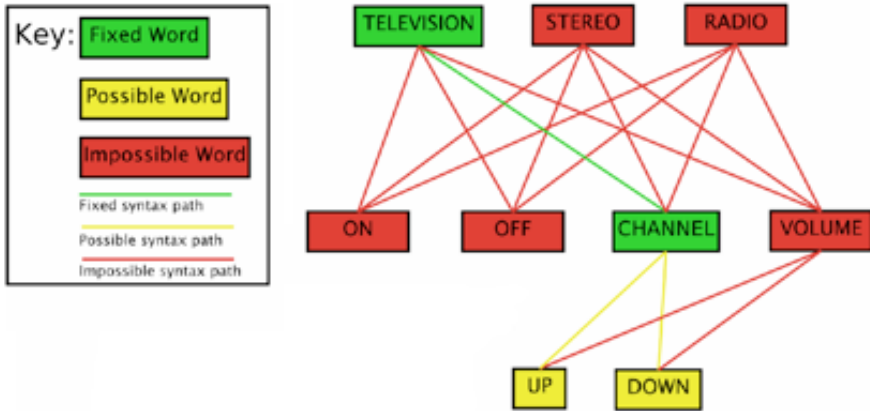
### 4.1 Controlled Testing: Single and Multiple Pass

Standard decoding tests consisted of speakers recording predefined sets of commands. The test sets are then decoded by both the Sphinx-4 base system and the multiple pass decoding system. The output sets are rated for correctness and the percent improvement, or lack thereof, is recorded. It should also be noted that speaker groups are kept representatively diverse, and all recording occurs under the same conditions. Table 1 presents the initial domain set considered for the tests.

Figure 4 presents an illustration of how our context based approach works in its multi-pass process. Thus far in our domain, multiple pass decoding has yielded measurable improvement over single pass decoding in all cases where improvement was possible. Below is an example test set recorded by a male, non-native English speaker. This case illustrates the potential of multiple pass decoding as presented in table 2. While standard Sphinx provided a 75% success rate, our multiple pass success rate was 93.75%.

**Table 1.** Initial domain used for command tests

Devices	Commands	Modifiers
Cabinet	on	lock
door	off	unlock
lights	open	up
radio	close(lock)	down
television	unlock	



**Fig. 4.** Illustration of possible syntax paths after second decoding pass

Although more tests are required, an overall 18.75 % improvement is promising. Furthermore our representative command set was spoken by a group of speakers containing both male and female, native and non-native speakers. The mean accuracy improvement of multiple pass decoding over all test sets was 7.81 %.

**4.2 Testing the Hypothesis**

After testing the approach proposed in the previous section, we test the behavior of the multiple pass; Figure 5 presents the performance as function of number of passes. Although the commands are typically short in length, the graph of mean accuracy level on each pass reveals a few important points. First, in the typical mean case, the simplest application of context driven decoding (a single additional pass) yields a noticeable (slightly over 6%) improvement. Second, the maximum accuracy achievable by this system is reached in fewer passes than there are words in each command. Although statistical validation is still required, the optimal number of passes must grow at a rate less than exponentially with respect to command length.

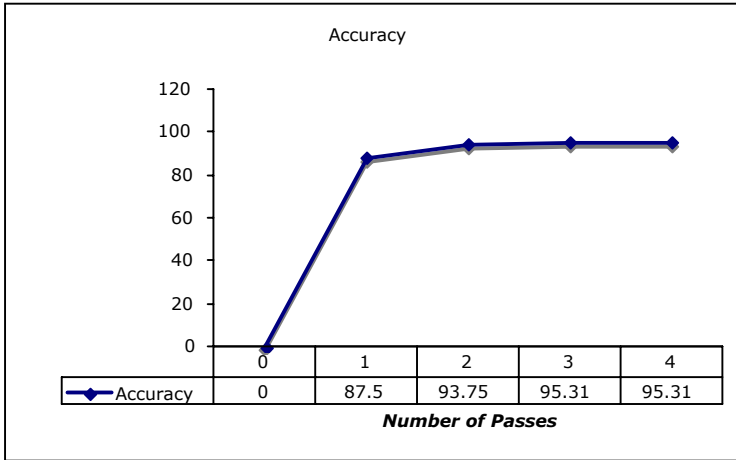


Fig. 5. Results of the context driven performance versus number of passes

### 4.3 Performance Metrics

The final consideration to address is a basic cost/benefit analysis of the proposed multiple pass enhancement to Sphinx. Here the metric to be considered is the time performance; in order to test this we define a time unit based on the original Sphinx algorithm. We define our time unit as the time for a command to be fully loaded, recognized and decoded spoken command in the original Sphinx; time is measured in milliseconds and compared to time of a single run to find length of time unit on particular system. Time for additional passes includes the time for the pass itself, plus time to reload updated parts of the recognizer.

Although the metric varies both with each command and with the number of passes, it is a useful measurement to determine the cost/benefit. The reasons for the variation stem from the fact that for Sphinx to decode a single command, the recognizer has to be loaded, then a decoding run must be made. If multiple passes are to be made, parts of the recognizer – though not the entire recognizer – must be reloaded, and another run made. Our metric can then be computed by measuring the time in the following processes:  $load + passes * decode + (passes - 1) * reload$ . One important observation is that additional passes require less time than the original pass, so time requirement increases linearly, though increments are never as great as the time for the first pass. Results obtained so far for a multiple speaker average command test show that a 96% accuracy is obtained in three passes at a cost of one and a half extra time; this is a reasonable cost performance ratio. The time unit used is based on the benchmark of a single decoding pass with the original Sphinx system; although measurements were taken in milliseconds for the system, they were converted to system independent time unit. In the best case, 96% accuracy was obtained in three passes consuming 2.06 units of time versus 1.0 for a single pass approach. A final fourth pass provided little enhancement with a total time of 2.59 time units. It must be said the Sphinx is a powerful system to start with, so our approach provided limited improvement to the system.

## 5 Final Remarks

Thus far in our domain, multiple pass decoding has yielded measurable improvement over single pass decoding in all cases where improvement was possible. Specifically, the context-awareness of multiple pass decoding prevented many obviously wrong commands, whereas the original Sphinx-4 system had no way to recognize obviously syntactically incorrect commands

Given the success of algorithmic multiple pass decoding within our limited domain, we are advised that the method should be optimized and fully tested in larger and more real domains, such as actual Smart Home. Furthermore there are issues yet to be addressed in our tests, some deal with the environment and some with the individuals. Concerning the environment, the distance of the speaker from the microphones and the quality of them; here it has been suggested that performance might be improved with the use of directional microphones or microphone arrays. Also the existence of background noise or multiple sources have to be taken into account, such as when the appliances are turned on. On the case of the individuals, it is not just the speaker accent, but the fact that in the case of smart homes, speaker age and speech abilities become important as the voice of older people may be more difficult to recognize. Aware of these issues, we will take them into consideration in order to achieve the desired goals of our research. Although in this case results may show a reduction in the overall performance, but it will be interesting to see if the context driven approach suggested here improves the overall performance, this will demonstrate the performance in a more realistic smart home environment, we expect to have new data to validate our approach.

Finally, as this requires more time-efficient multiple pass decoding, our immediate goals include improving integration of our algorithm into the Sphinx system, as well as developing more efficient data structures to store syntax maps. In addition, we aim to expand our base of test users to better judge how effective our algorithms are for the general user.

## References

1. Ravishankar, M.K.: Efficient Algorithms for Speech Recognition. Ph.D Thesis, Carnegie Mellon University, Tech Report. CMU-CS-96-143 (May 1996)
2. Harper, R. (ed.): Inside Smart Home. Springer, Berlin (2003)
3. Wang, C., Chung, G., Seneff, S.: Automatic Induction of Language Model Data for A Spoken Dialogue System. Special Issue of the Springer Journal on Language Resources and Evaluation 40(1), 25–46 (2006)
4. Lamere, P.K., Walker, W., Gouvea, E., Singh, R., Raj, B., Wolf, P.: Sphinx-4: A Flexible Open Source Framework for Speech Recognition Sun Microsystems, Report Number: TR-2004-139
5. Lamere, P.K., Walker, W., Gouvea, E., Singh, R., Raj, B., Wolf, P.: Design of the CMU Sphinx-4 decoder. In: Proceedings of the 8th European Conference on Speech Communication and Technology, Geneva, Switzerland, pp. 1181–1184 (September 2003)
6. Huggins-Daines, D., et al.: PocketSphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices. In: Proc. of ICASSP 2006, Toulouse (May 16-19, 2006)

7. Walker, W., Lamere, P., Kwok, P.: FreeTTS - A Performance Case Study, SUN (2002)
8. Intille, S.S.: The goal: smart people, not smart homes. In: Proceedings of the International Conference on Smart Homes and Health Telematics, IOS Press, Amsterdam (2006)
9. Siivola, V., Pellom, B.L.: Growing an n-gram language model. In: INTERSPEECH-2005, pp. 1309–1312 (2005)
10. Wang, W., Stolcke, A., Harper, M.P.: The Use Of A Linguistically Motivated Language Model In Conversational Speech Recognition. In: Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing, Montreal, Canada (2004)
11. Lee, K.F., Hon, H.W., Reddy, R.: An overview of the SPHINX speech recognition system. *IEEE Transactions on Acoustics, Speech and Signal Processing* 38(1), 35–45 (1990)
12. Huang, F.A., Hon, H.W., Hwang, M.Y., Rosenfeld, R.: The SPHINX-II speech recognition system: an overview. *Computer Speech and Language* 7(2), 137–148 (1993)

# Emotion Estimation Algorithm Based on Interpersonal Emotion Included in Emotional Dialogue Sentences

Kazuyuki Matsumoto<sup>1</sup>, Fuji Ren<sup>1,2</sup>, Shingo Kuroiwa<sup>1</sup>, and Seiji Tsuchiya<sup>1</sup>

<sup>1</sup> Faculty of Engineering, The University of Tokushima,  
Tokushimashi 770-8506, Japan

<sup>2</sup> Beijing University of Posts and Telecommunications  
Beijing, 100876, China

{matumoto, ren, kuroiwa, tsuchiya} @is.tokushima-u.ac.jp

**Abstract.** Emotion recognition aims to make computer understand ambiguous information of human emotion. Recently, research of emotion recognition is actively progressing in various fields such as natural language processing, speech signal processing, image data processing or brain wave analysis. We propose a method to recognize emotion in dialogue text by using originally created Emotion Word Dictionary. The words in the dictionary are weighted according to the occurrence rates in the existing emotion expression dictionary. We also propose a method to judge the object of emotion and emotion expressivity in dialogue sentences. The experiment using 1,190 sentences proved about 80% accuracy.

## 1 Introduction

The research of emotion recognition by computer has progressed in the various fields such as artificial intelligence, speech recognition, natural language processing and image processing. Human emotion recognition by computer is expected to realize creating a humanoid robot that people can communicate without unease. Our research group has been engaged in development of emotion recognition technology for “Affective Interface” which can communicate with people by using language, voice sound, facial expression and gesture [1], [2], [3]. Research of emotion recognition by natural language is basically divided into two kinds. One is for extracting emotional expressions and the other is for estimating emotions of a speaker from utterance. Some of the research on extracting emotional expressions focus on judging semantic orientation of a sentence or extracting words based on positive or negative index [4], [5]. The representative research of estimating human emotions from dialogue text are [6], [7] and [8] that primarily focus on words used in sentences. In conversational context, an emotion expressed in utterance does not always correspond to a speaker’s emotion. It should be considered whose emotion it is. For example,

- (A) “*Kimi wa totemo ureshisoudane.*” ( “You look very happy.” )
- (B) “*Tottemo ureshii.*” ( “I’m very happy.” )

- (C) “*Kare wa totemo ureshisouda.*” ( “He looks very happy.” )
- (D) “*Kare wa totemo kashikoi.*” ( “He is very smart.” )

In (A) “happy” is used to describe a speaker’s impression about someone talking with (conversation partner) or other person. In this case emotional description “happy” is not corresponding to the speaker’s emotion. Usually when a speaker describes about his/her own emotion by using the first person as subject or just omitting subject, the expressed emotion (“happy” in (B)) is interpreted as a speaker’s emotion. An emotional state of the third party is expressed in (C) and an evaluation to the third party is described in (D) in the eyes of a speaker. These examples suggest the necessity of considering whom the emotion expressed in an utterance is directed to and whom that emotion belongs to.

[9] divides emotion into two kinds: “Self-emotion” and “Interpersonal-emotion” The emotions classified according to interpersonality and stability of emotion are shown in Table 1. “Self-emotion” is generated in describing speaker’s own emotion, while “Interpersonal-emotion” is generated in describing other person’s emotion. One of the problems of existing methods is that they recognize

**Table 1.** Emotion Classification (excerpted from [9])

Class Name	Stability	Interpersonal / Self	Emotion Categories
Impulse	No	Self	Sorrow, Anger, Joy, etc.
Mood	Yes	Self	Anxiety, Happy, Proud, Loneliness, etc.
Reaction	No	Interpersonal	Surprise, Shame, Chagrin, Fear, etc.
Attitude	Yes	Interpersonal	Love, Respect, Hate, etc.

expressed emotion as the speaker’s emotion because they do not consider subject and object of an emotion. In this paper, we create an emotionally weighted dictionary “Emotion Word Dictionary” based on a corpus collected emotional expressions categorized for each emotion. We also propose a method to estimate a speaker’s emotion by identifying if the expressed emotion is the speaker’s emotion or not then construct an emotion estimation system based on the proposed method. We focused on specifying who the expressed emotion belongs to: to speaker him/herself or to others by identifying whether the sentence emotion corresponds to the speaker’s emotion. If the estimated emotion belongs to other person other than a speaker, the system converts the emotion into the emotion of a speaker based on an emotion estimation rule and outputs.

The outline of this paper is as follows. In section 2 we describe the construction of “Emotion Word Dictionary” and an equation calculating the level of emotional expression (EEL-equation). In section 3 we propose a method to estimate emotion considering interpersonality of the emotion. The section 4 describes the system construction. In section 5, the proposed method is evaluated by the system. Finally, in section 6 concluding remarks are made and future works discussed.



## 2 Constructing Emotion Word Dictionary

In this research we created an “Emotion Word Dictionary” which consists of content words taken from emotion expressing sentences mainly in [10] and weighted according to the importance of the words in each emotion category. The word is the minimum unit for human to recognize/express the emotion, therefore we annotated the emotion in word level. Total number of emotion expressing word in the “Emotion Word Dictionary” was 15,218. Emotion categories were decided based on distributions of the emotions expressed in the collected sentences.

The 8 basic emotion categories are: “joy,” “anger,” “anxiety,” “sorrow,” “surprise,” “love,” “hate” and “respect.” The following emotion subcategories are also selected for each 8 emotion category. For example, Affective Task of SemEval [11] defined six categories (anger, disgust, fear, joy, sadness, surprise) as basic emotions for text-based emotion recognition. As their categories were designed for headlines of news articles, they did not cover subjective emotions. Due to our research aim of emotion recognition of a speaker, we added two subjective emotions of “love” and “respect” to their six categories. “Anxiety” in our category corresponds to their category of “fear.” The state without any specific emotion occurred was defined as “neutral.”

**Table 2.** Basic Emotion Category

Category	Subcategory
Joy	Pleasure, Happy, Appreciation, Funniness, Anticipation, Proud
Anger	Resentment, Spite, Accusation
Sorrow	Loneliness, Depression, Regret, Chagrin, Compassion, Despair
Surprise	Trouble, Shocked
Hate	Contempt, Jealousy, Hatred, Complaint
Respect	Adoration, Approbation, Admiration, Longing, Nostalgia, Envy
Anxiety	Fear, Quizzicalness, Worry
Love	Like, Reception, Interest
Neutral	

The importance  $W(E_i, w_j)$  of the word  $w_j$  in emotion category  $E_i$  is calculated following the same idea of the conventional method of *tfidf* value [12], which is usually used for information retrieval. The value  $\alpha$  in Equation 1 shows normalized coefficient and is calculated based on Equation 2. The value  $l$  indicates the total number of unique word.  $freq(w_j, E_i)$  indicates the frequency of  $w_i$  in sentence assembly classified into  $E_i$ .  $cf(w_j)$  indicates the number of emotion category in which  $w_j$  is included.  $N$  shows the total number of emotion category. Words with semantic attributes unrelated to emotion are removed in advance.

$$W(w_j, E_i) = \alpha * freq(w_j, E_i) * \log \frac{N}{cf(w_j)} \tag{1}$$

$$\alpha = \frac{1}{\sqrt{\sum_{m=1}^l freq(w_m, E_i)^2 * \frac{1}{cf(w_m)^2}}} \tag{2}$$

Example of Emotion Word Dictionary is shown in the Figure 1.

```

<EmotionWordDictionary>
  <word name="eerie">
    <emotionWeight>
      <emotion category="anxiety" weight="15.6924" />
    </emotionWeight>
  </word>

  <word name="awful">
    <emotionWeight>
      <emotion category="fear" weight="21.1686" />
      <emotion category="sorrow" weight="0.5688" />
      <emotion category="excitement" weight="0.5304" />
    </emotionWeight>
  </word>

```

Fig. 1. Example of Emotion Word Dictionary (XML Format)

### 2.1 Emotion Estimation Method Based on Emotion Word Dictionary

This section explains an emotion estimation method based on word weight assigned to each emotion category. We regard the method as a baseline of emotion estimation method. Emotional level for each emotion category was calculated using Equation 3. We defined Equation 3 as "Emotion Expression Level - equation".  $EP(S, E_x)$  indicates emotional level of emotion ( $E_x$ ) in sentence  $S$ .  $W(w_i, E_x)$  indicates weight of word ( $w_i$ ) in sentence  $S$  for emotion category ( $E_x$ ). In addition,  $Neg(w_i)$  is function to return '-1' if the word coming after  $w_i$  is negative word and to return '1' if the word coming after  $w_i$  is not negative word (Equation 4).

$$EP(S, E_x) = \sum_{i=0}^n W(w_i, E_x) * Neg(w_i) \tag{3}$$

$$Neg(w_i) = \begin{cases} -1 & \text{if } w_{i+1} = \text{Negative Word} \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

Finally, two of the highest emotion categories of  $EP(S, E_x)$  are outputted as estimated emotions of sentence  $S$ .

### 3 Emotion Expression Judgment by Detecting Person Referent Word

In the previous section we proposed a method to estimate emotion based on emotion expressing words included in sentences by using EEL-equation. However, emotion of word does not always correspond to speaker’s emotion. Object of the emotion expression may be important to be considered. In this paper, we defined “personal pronoun” or “proper noun of personal name” as person referent word and considered that these words were the key of judging object and subject of emotion expression.

At the first step we created a database named “Person Emotion Expression Database (PEDB).” The database consists of three elements extracted from assembly of emotion tagged sentences: 1) pairs of person referent and postpositions before/after the person referent, 2) information about if the words before/after the pairs contain emotion or not and 3) probability of the sentence having emotion of a speaker or not.

The person referent words consists of words extracted from a thesaurus “A Japanese Lexicon.” [13] Part of speech (“proper noun-person name”) was referred to identify proper noun such as ”personal name.” Whether emotion expression of a speaker exists or not is judged based on this database. If sentence emotion is judged as ”neutral” based on PEDB, the expressed emotion in the sentence is judged as not expressing the speaker’s emotion. If an emotion is judged as emotions other than “neutral,” the expressed emotion in the sentence is judged as the speaker’s emotion. Table 3 shows the example data. Pairs of person referent and postpositions before/after the person referent of the inputted sentence are compared with that of database. Equation 5 is used to calculate the cosine similarity ‘*SimScore*.’ If the *SimScore* is over the threshold value, obtain the pattern of emotion expression. If we get multiple expression patterns, average the scores to judge the pattern of emotion expression.  $PV(X)$  indicates vector of person referent and postpositions extracted from sentence  $X$ .  $PV(Y)$  indicates vector of person referent and postpositions in “Person Emotion Expression Database”.  $|PV(X)|$  and  $|PV(Y)|$  indicate size of each vector.

$$SimScore = \frac{PV(X)PV(Y)}{|PV(X)||PV(Y)|} \tag{5}$$

We focused on co-occurrence of person referent word and part of speech of emotion word such as adjective (*EAdj*) or verb (*EVerb*) as one of the important

**Table 3.** Example of Person Emotion Expression Database

Expression	Before	After	neutral:other
[Third person]- <i>wa</i> / [Second person]- <i>ga</i>	0 / 1	0 / 0	0.55 : 0.45
[Second person]- <i>ni</i>	1	0	0.10 : 0.90
[Person name]- <i>wa</i>	1	0	0.60 : 0.40

**Table 4.** Co-occurrence Rule

Pattern	Subject of Emotion
[Other] + {[ <i>wa</i> ] or [ <i>ga</i> ]} + [EAdj]	$E_{sp}$ (interpersonal)
[EAdj] + [Other]	$E_{sp}$ (interpersonal)
[Other] + {[ <i>wa</i> ] or [ <i>ga</i> ]} + [EVerb]	$E_{ot}$ (self)
[Speaker] + {[ <i>wa</i> ] or [ <i>ga</i> ]} + [EAdj]	$E_{sp}$ (self)

factors causing speaker’s emotion. We defined rules shown in Table 4. The proposed method judges whether emotion expressions in sentence are speaker’s or not by using “Person Emotional Expression Database” and co-occurrence rule shown in Table 4.

### 3.1 Process of Personal Relationship Extraction

We decided a rule to estimate other person’s emotion when person referent indicating except a speaker is used near emotional expression. However, to interpret other person’s emotion from a viewpoint of a speaker then convert other person’s emotion to emotion on speaker’s side, it would be necessary to consider attributes of person referent word. For example, in sentences: “*Kimi wa erai ne.* (How admirable you are!)” and “*Anata wa erai desune.* (You are commendable.)” attitude of a speaker to the emotional object is different. In Japanese “*Kimi*” is used for someone in equal or lower position, while “*Anata*” is usually used for person in higher position.

For that reason, we thought that it is necessary to extract the personal relation between speaker and the object indicated by person referent. Concretely, relationships that person referent indicates can be categorized into the following three types. By combining those three, we decided the hierarchical relationships between a speaker and conversation partner or other persons.

- “First person” indicates relationship between a speaker and object of a conversation.
- “Second person” indicates relationship between a speaker and object of a conversation.
- “Third person” indicates relationship between a speaker and others.

We do not treat person referent word of “proper noun - person name” in this paper because it requires more detailed data about interpersonal relationship. The hierarchical relation between a speaker and other person will give a clue to estimate whether a speaker tend to empathize with other person or not. We calculated the hierarchical relation between a speaker and other person by Equation 6. If *RankFlag* is 1, other person (*Ot*) is in lower rank than the speaker (*Sp*). If *RankFlag* is 0, other person (*Ot*) is in equal rank to the speaker (*Sp*). If *RankFlag* is -1, other person (*Ot*) is in higher rank than the speaker (*Sp*). And, if *RankFlag* is 2, the relation between speaker and other person is unknown.

$$RankFlag = \begin{cases} 1 : & \text{if } Position(Sp) > Position(Ot) \\ -1 : & \text{if } Position(Sp) < Position(Ot) \\ 0 : & \text{if } Position(Sp) = Position(Ot) \\ 2 : & \text{unknown} \end{cases} \quad (6)$$

The dictionary registered the relations of *RankFlag* and personal referent words was created as the ‘‘Personal Referent Word Dictionary.’’ If an emotion of other person is extracted, output the emotion of a speaker by applying ‘‘Empathy Rule’’ shown in the Figure 2.

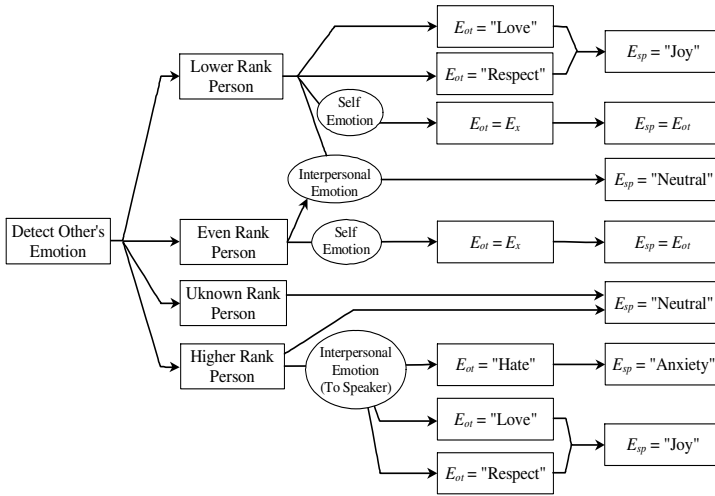


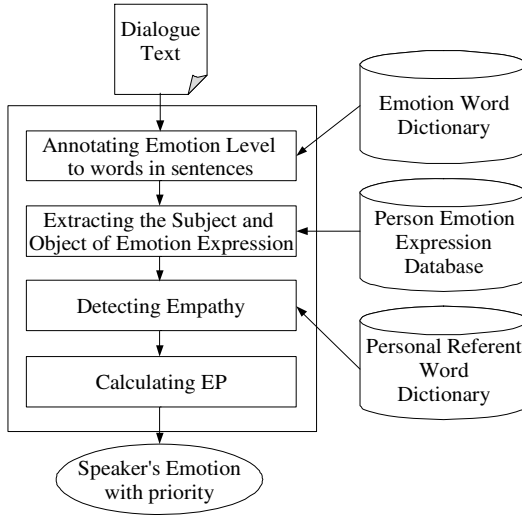
Fig. 2. Example of Empathy Rule

### 3.2 Experiment for Judging Speaker’s Emotion

An evaluation experiment (closed test) was conducted to determine if the sentence had a speaker’s emotion or not by using the ‘‘PEDB.’’ In particular, the system judged whether inputted sentence expressed the speaker’s emotion or not by identifying person in the sentence. For experiment, we used two test collections: Test collection A (1,190 sentences of utterance) and test collection B (1,774 sentences of dialogue taken from acting script). Success rate was 66.0% in test collection A and 78.7% in test collection B. Higher success rate in test collection B is probably because that test collection B is based on acting script with more explicit emotional expressions.

## 4 Construction of a Prototype System Based on the Proposed Method

Section 4 suggests an emotion estimation system based on the proposed methods. Figure 3 shows architecture of the system. First, inputted sentence is divided into



**Fig. 3.** Architecture of Emotion Estimation System

morphemes by using Japanese morphological analyzer “ChaSen” [14], and morphemes expressing emotions in sentence are annotated emotion expression level by referring “Emotion Word Dictionary”. Second, system extracts expressions concerning person and judges if the speaker’s emotion is “neutral” or “other” by referring to “Person Emotion Expression Database.” When the speaker’s emotion is judged as “other,” then the tagged emotional expressions are considered to be the speaker’s emotion. Third, when the speaker’s emotion is judged as “neutral” and the object of emotion expression is other person, “Empathy Rule” is applied. Fourth, system calculates emotion expression level of a speaker based on EEL-equation. Finally, the two highest levels of emotion categories are outputted as speaker’s emotions.

## 5 Evaluation of the Proposed Method

To confirm the effectiveness of the method for estimating emotion considering person identification, experiment was conducted with the proposed prototype system in the previous section.

### 5.1 Experimental Method

For experimentation, the corpus of 1,190 sentences tagged with 9 emotions of a speaker were used, which was the same corpus used for creating PEDB. First, we used EEL-equation then applied Empathy Rule for emotion estimation. After that, open test was also conducted by using the smaller corpus of 90 sentences tagged with 9 emotions of a speaker, which was not used for PEDB creation.

These test collections are tagged by one person allowing multiple tagging for each sentence. The evaluation was made by comparing the emotion categories outputted by the system with the manually tagged emotions. When the two of the highest emotions estimated by the system correspond to the manually tagged

**Table 5.** Result of Evaluation(closed, open)

[i] open test (only EEL)

Category	Precision	Recall	F-Measure
Anger	87.5% ( 14 / 16 )	35.9% ( 14 / 39 )	50.9%
Joy	91.4% ( 53 / 58 )	48.6% ( 53 / 109 )	63.5%
Love	100.0% ( 3 / 3 )	13.6% ( 3 / 22 )	24.0%
Surprise	93.8% ( 15 / 16 )	57.7% ( 15 / 26 )	71.4%
Sorrow	89.7% ( 35 / 39 )	42.2% ( 35 / 83 )	57.4%
Hate	90.9% ( 10 / 11 )	4.7% ( 10 / 212 )	9.0%
Anxiety	60.0% ( 3 / 5 )	5.0% ( 3 / 60 )	9.2%
Respect	100.0% ( 4 / 4 )	4.6% ( 4 / 87 )	8.8%
Neutral	75.0% ( 6 / 8 )	1.0% ( 6 / 589 )	2.0%
Average	89.4% (143 / 160)	11.7% ( 43 / 1227 )	20.6%

[ii] closed test (EEL and Empathy Rule)

Category	Precision	Recall	F-Measure
Anger	87.0% ( 20 / 23 )	51.3% ( 20 / 39 )	64.5%
Joy	74.0% ( 57 / 77 )	52.3% ( 57 / 109 )	61.3%
Love	92.3% ( 12 / 13 )	54.6% ( 12 / 22 )	68.6%
Surprise	88.5% ( 23 / 26 )	88.5% ( 23 / 26 )	88.5%
Sorrow	73.4% ( 47 / 64 )	56.6% ( 47 / 83 )	64.0%
Hate	90.1% (118 / 131)	55.7% (118 / 212)	68.8%
Anxiety	58.3% ( 14 / 24 )	23.3% ( 14 / 60 )	33.3%
Respect	53.7% ( 22 / 41 )	25.3% ( 22 / 87 )	34.4%
Neutral	92.3% (482 / 522)	81.8% (482/589)	86.8%
Average	86.3% (795 / 921)	64.8% (795 / 1227)	74.0%

[iii] open test (EEL and Empathy Rule)

Category	Precision	Recall	F-Measure
Anger	100.0% ( 4 / 4 )	40.0% ( 4 / 10 )	57.1%
Joy	100.0% (10 / 10)	100.0% (10 / 10)	100.0%
Love	80.0% ( 4 / 5 )	40.0% ( 4 / 10 )	53.3%
Surprise	100.0% (10 / 10)	100.0% (10 / 10)	100.0%
Sorrow	50.0% ( 4 / 8 )	40.0% ( 4 / 10 )	44.4%
Hate	80.0% ( 4 / 5 )	40.0% ( 4 / 10 )	53.3%
Anxiety	0.0% ( 0 / 1 )	0.0% ( 0 / 10 )	0.0%
Respect	100.0% ( 2 / 2 )	20.0% ( 2 / 10 )	33.3%
Neutral	83.3% ( 5 / 6 )	50.0% ( 5 / 10 )	62.5%
Average	84.3% (43 / 51)	47.8% (43 / 90)	61.0%

emotions, then the estimation was judged correct then Precision, Recall and F-Measure (Equation 7) were calculated. Results are shown in Table 5. [i] shows the results of the open test with EEL-equation for 1,190 sentences, [ii] and [iii] shows the results of the open/closed test with EEL-equation and Empathy Rule.

$$F - Measure = \frac{2 * precision * recall}{precision + recall} \quad (7)$$

## 5.2 Discussion

The experimental results indicated that the precision was over 80% on average at open test. The precision of closed test decreased from that with EEL-equation, however, recall rate increased about 50%. The primary factor of these results is that recall of “neutral” increased by judging the emotion expressivity based on person referent word. Consequently, the validity of the proposed method was confirmed. The problem was some sentences were failed in estimation because the relationship between a speaker and other person was not judged correctly owing to lack of consideration of modality. However, in the sentences which have error, Anxiety was low in precision because emotion of anxiety was often described without explicit emotional expressions. To consider implicit emotion expressions, it would be necessary to consider sentence-final expressions in addition to words.

## 6 Conclusion

In this paper, we proposed an emotion estimation method considering emotional weight of word and interpersonal relationship expected from sentence. We also constructed a system based on the proposed methods and conducted experiments estimating emotion from sentences with emotional expressions. The results showed approximately 80% of accuracy on average and proved enough effectiveness of the proposed methods. As the recall was rather low (approx. 40%) at open test, the following items would be necessary to improve:

1. Expand emotion word dictionary and reconstruct Person Emotion Expression Database.
2. Consider words other than emotional words in a sentence.

Modality also should be considered in converting other person’s emotion to an emotion on a speaker’s side for the purpose of improving precision of emotion estimation.

## Acknowledgment

This research has been partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (B), 19300029, 17300065, Exploratory Research, 17656128.



## References

1. Ren, F.: Recognizing Human Emotion and Creating Machine Emotion. Invited Paper, Information, Japanese 8(1), 7–20 (2005)
2. Matsumoto, K., Mishina, K., Ren, F., Kuroiwa, S.: Emotion Estimation Algorithm based on Emotion Occurrence Sentence Pattern. *Journal of Natural Language Processing* 14(3), 239–271 (2007)
3. Minato, J., Bracewell, D.B., Ren, F., Kuroiwa, S.: Statistical Analysis of a Japanese Emotion Corpus for Natural Language Processing. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) *ICIC 2006. LNCS (LNAI)*, vol. 4114, pp. 924–928. Springer, Heidelberg (2006)
4. Takamura, H., Inui, T., Okumura, M.: Extracting semantic orientations of words using spin model. In: *Proceedings 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp.133–140 (2005)
5. Turney, P.D.: Thumbs up? thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 417–424 (2002)
6. Tokuhisa, M., Murakami, J., Ikehara, S.: Construction of Text-dialog Corpus with Emotion Tags Focusing on Facial Expression in Comics. *Journal of Natural Language Processing* 14(3), 193–218 (2007)
7. Tsuchiya, S., Yoshimura, E., Watabe, H., Kawaoka, T.: The Method of the Emotion Judgment Based on an Association Mechanism. *Journal of Natural Language Processing* 14(3), 219–238 (2007)
8. Mera, K., Ichimura, T., Yamashita, T.: Complicated Emotion Allocating Method based on Emotional Eliciting Condition Theory. *Journal of the Biomedical Fuzzy Systems and Human Sciences* 9(1), 1–10 (2003)
9. Hirota, K.: *Technology Affecting Emotion and Science Recognizing Feeling*, in Japanese, Shinpu-sha (2001)
10. Nakamura, A.: *Emotion Expression Dictionary*, in Japanese, Tokyodo (1993)
11. Affective Text SemEval Task 14 (2007), <http://www.cse.unt.edu/~rada/affectivetext/>
12. Salton, G., Buckley, C.: Term weighting approaches in automatic retrieval. *Information Processing and Management* 24, 513–523 (1988)
13. Ikehara, S., Miyazaki, M., Shirai, S., et al.: *A Japanese Lexicon*, Iwanami Shoten (1999)
14. Matsumoto, Y., Kitauchi, A., Yamashita, T., Hirano, Y.: *Japanese Morphological Analysis System ChaSen*, NAIST (2002)

# The Framework of Mental State Transition Analysis

Peilin Jiang<sup>1,2</sup>, Hua Xiang<sup>1</sup>, Fuji Ren<sup>1</sup>, Shingo Kuroiwa<sup>1</sup>, and Nanning Zheng<sup>2</sup>

<sup>1</sup> The University of Tokushima, Tokushima, Japan  
jiang@is.tokushima-u.ac.jp

<sup>2</sup> Xi'an Jiaotong University, Xi'an, 710049, China

**Abstract.** The Human Computer Interaction (HCI) Technology has emerged in the different fields in applications in computer vision and recognition systems such as virtual environment, video games, e-business and multimedia management. In this paper we propose a framework of designing the Mental State Transition (MST) of a human being or virtual character. The expressions of human emotion can be easily remarked by facial expressions, gestures, sound and other visual characteristics. But the potential MST modeling in affective data are always hidden actually. We analysis the framework of MST and employ DBNs to construct the MST networks and finally the experiment has been implemented to derive the ground truth of the data and verify the effectiveness.

**Keywords:** Mental State Transition, HCI, Psychological Experiment, Virtual Character.

## 1 Introduction

The affective computing becomes pervasive and crucial in the Human Computer Interaction with the exploit to the psychology and neuroscience field and many multidisciplinary fields. Person whose careers are full of different computers in both business and everyday life are attending to communicate with the 'them' more directly and effectively. More humanity in HCI will bring more comprehensive communications.

Normally, there are two different approaches to explore machine-style affective property and mental states. One is from the neuroscience theoretically and the other is based on psychological experiment. Only from the engineering method we can hardly derive the performances of the practical movement in mind. Actually there are the sensors of the human being such as eyes, mouth and ears through which people obtain the information and responses are performed by facial expressions, words and speeches.

The motivation of analysis of human mental state is not only to comprehend the mechanism of the humanity, actually by now the existing knowledge is not enough for us to describe the human mental states, but also to realize the machine-style emotions that can make the computer to listen and speak more like a human being.

There are several problems in machine-style emotions. For example, the emotion recognition, emotion synthesis and emotion state (mental state) transitions. In the first two aspects, there are various categories of features like facial expression, voices and speech. Many related researches and implementations in these aspects are developed in the decades. And also for these ones, the neuroscience methods and the approaches in computer vision, speech recognition and natural language processing have been applied.

Conversely, for the last one, mental state transition, more previous methods are based on psychological assumption and empirical data. The efforts have been firstly done from the neuroscience etc. No direct evidences has been verified that the detailed explanation of the human mental state is correctly promising. The common hypothesis is that the human mental states are independent and can be understood by observation features. The psychological experiments have been done that there at least seven mental states including neural one [2].

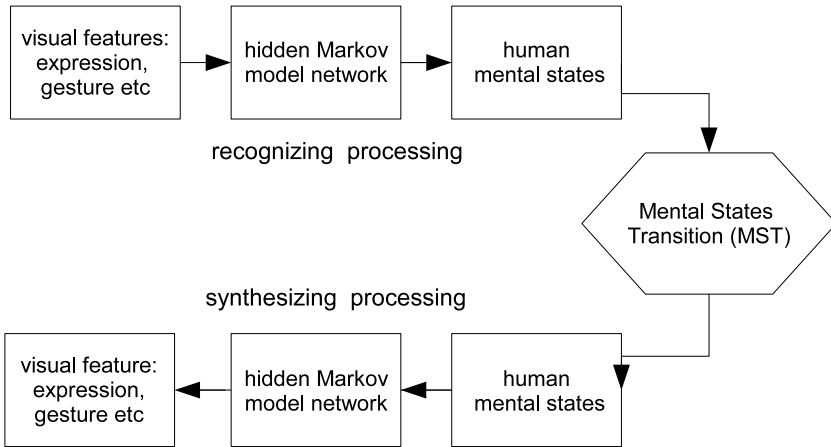
However, the first task before recognizing and synthesis of human emotions and affective computing is comprehending the status of the mental state transition. After that, we can simulate the analogous human MST in computers. In this paper, we first describe a framework of the recognition processing of the mental state, then the MST processing and the synthesis processing in next phase. To understand clearly, we propose a psychological experiment of MST modeling and a application of the virtual character of human-computer interaction.

In Section 2 we analyze the whole framework of the mental state transition procedure including mental states recognizing, MST and the synthesis processing. Section 3 describes the approaches we employed in the framework, basically DBNs in the architecture of MST modeling and some previous researches. In section 4, we describe the psychological experiment on MST modeling and the results analysis. In section 5, we introduce the sample of HCI application with a virtual character in MST. The final section remarks the conclusion and the future work.

## 2 Mental State Analysis

Generally, we assume the human mental states are independent and transform between each other under different surrounding situations. But this is not the direct impression because we can not observe any mental states and the change of them. Fortunately, the human mental state can only be observed by visual characters such as facial expressions, voice or words. These observations are corresponding to the states of human mind probabilistically. For an instance, it is more probable to express a smile or laugh when the user is in the mental state of happiness. We can describe the happiness of human mental state with the observations like smiling expression and laughing although we know these performances or gestures are not caused by the happiness tentatively.

Similarly, the transition movements between the different mental states are invisible and this make it hard to discover the performances of these inside properties. Hence, the current approaches on research of mental states transition are based on psychological experiments. The previous works exploring the categories



**Fig. 1.** Architecture of the Framework of Mental State Transition and Analysis

of human mental states are reasonable and the transitions between them cause the transformations of the observations of visual features.

The synthesis from the mental state transition analysis to observations is also the statistical processing of the temporal finite states transition network. However, this procedure synthesizes the outside performances of a people. The whole architecture can be introduced as the following Fig.1.

### 3 Structure of Mental State Transition

As showing in the Fig.1, in our work we used the probability transition network models in these processing such as HMM and MST modeling. Previously, there are several researches in these aspects.

#### 3.1 Related Works

The hidden Markov model has been applied in analysis of the human expressions and mental states before. Ira Cohen and T.S Huang et al [6] used a multilevel hidden Markov model in research of emotion recognition from facial expressions. the motivations they proposed are the segmenting and recognizing human facial expressions from video sequences automatically. Hence, the multilevel HMM architecture had been employed and comparing with a single HMM method. Six mental states has been considered in their work except the neural state and all of transitions have to go through the neural one and be described by a high-level HMM. However, it is hardly to obtain reliable affective data through their models.

In a case of implementation, Xiangyang Li et al [5] used a DBNs model in affective state detection and user assistance work. In their research, the observations concluded eye movement, gaze, facial expression, hand and head gesture

etc. Similarly the affective state (mental state in our paper) is represented by hidden nodes in DBNs. And mathematically, the affective entropy approach is used to present the affective states. The detection from the observable data helped to make the decision on assistances. The framework is practical and reasoning in their cases. But it is mostly focused on the relationship between the observations and affective states. But there is few exploring on the mechanism of the mental transition.

### 3.2 Structures in Mental State Analysis

Generally, the Bayes Networks (BNs) are some kind of probabilistic graphical models representing joint probabilities conditions of a collection of random variables. The basic kind of DBs are Static Bayes Networks that are worked on one time instant only and it is useless in temporal system modeling. In general, the Dynamic Bayes Networks (DBNs) are composed by a series of time slices of static BNs. Kalman filtering and HMMs are the unique representative models of DBNs.

We decided to propose this hierarchical framework of DBNs into our work because that, first the dynamic procedure of the transitions of mental states under changing surrounding environments are just the system modeling suitable for the DBNs. Then, the architecture also allows to predict of the incoming events. Also, the hierarchical structure supports information from different levels.

As showing in Fig.1, we will apply the hidden Markov models in the recognizing processing and synthesizing processing respectively because we can represent the mental state with the hidden state variables and visual features with evidence variables as we introduction before in Section 2.

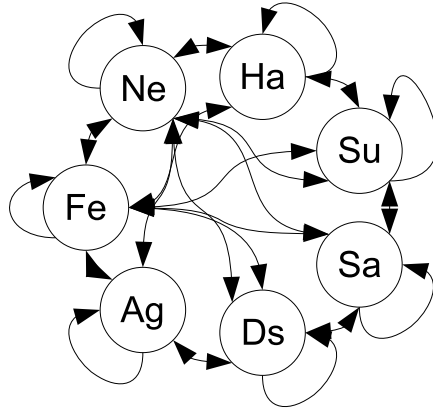
In the MST processing, the BNs is used and the generic framework we used Mental State Transition modeling [7] is showed as in Fig.2. The mental states of human being are influenced by the most closed states and the surrounding situations. However, the information of individual user also are considered as the conditions. In order to obtain the mathematical description we have done the psychological experiment for MST.

## 4 Experiment on MST

### 4.1 Experiment Design

Though, various emotional models have been proposed in previous studies (the Plutchik's Multidimensional Model [3], the Circumplex Model of Affect etc. [4]), there are few studies in which describe the mental situation appropriately in a numerical way that can be simulated in a computer directly.

For this reason, a MST model, which can be realized by engineering approach, should necessarily be created. Picard pointed out that a model such as the HMM can be used not only to recognize certain affective patterns, but also to predict what state a person is most likely to be in next, given the state they are in [1] now.



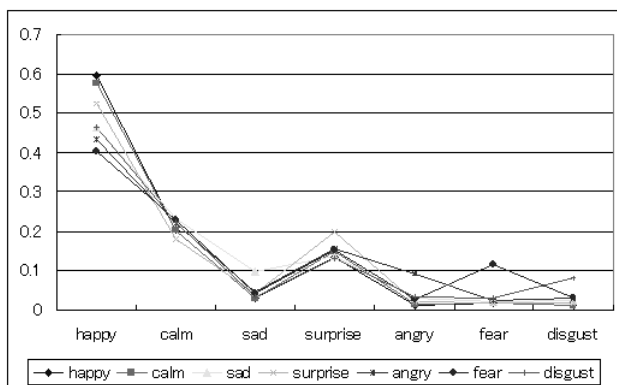
**Fig. 2.** Mental State Transition modeling. Ne: neural, Ha: happy, Fe: fear, Su: surprise, Sa: sad, Ds: disgust, Ag: angry. Nodes represent mental states and linking arcs represent the causal relations (we have not displayed all links in order to look clearly, actually all nodes are linked by arcs).

In our research, we presume that human emotional state is made up by seven basic emotional states (happy, sad, surprise, angry, fear, disgust and neutral) [2]. The six archetypal emotions are all full-blown emotions. Then, these seven discrete emotional states can construct an emotional space as Fig.2 shows.

## 4.2 Experiment Procedure

The psychological experiment [7] required participants to fill out a table which was designed for creating transitions among seven basic emotional states. In the experiment some sentences [8] depicting the situations of different stimulations are given to the participants. For instance under happy condition, the hint sentences are: please imagine you are in a certain emotional state (ex. imagine that you are angry at current time), at this moment if a happy thing is taken place (ex. you have passed an important exam or have obtained promotion etc.), what emotional state you would be at next period of time.

The aim of the psychological experiments is to detect and describe human emotional states transition by subjective Language questionnaire. We use the Conditional Probability Table(CPTs) as the foundation of the model of Mental State Transition. In our experiments, CPTs are obtained through psychological questionnaires. In the experiment, we had about 200 participants recruited primarily from different high schools and universities in China and Japan respectively. The ages of the participants ranged from 16 to 30 years old. 112 of them were males and 88 were females. The questionnaires were filled in in a classroom setting. Each of the participants was required to fill out the questionnaire giving serious thinking about emotional mental state transition.



**Fig. 3.** The transition probabilities under happy stimuli

After data normalization, the unbiased estimated means are calculated to evaluate the CPTs of the model. From these tables, the unbiased mean value of each emotional state of the network under various conditions has been calculated. And we can predict the transition procedure of emotional mental states in various situations.

As studying CPTs, we can find some general constraints of these mental states transition situations under various conditions. By comparing the graphs of CPTs we found that transition probabilities under neutral stimuli is absolutely different from the others archetypal emotion stimuli. The broken lines in Fig. 3 under the other six archetypal emotional stimuli almost have same tendency. In table 1, several general constraints of these mental states transition situations under neutral stimuli can be clearly concluded [9].

### 4.3 Result Analysis

Firstly, under neutral condition, the previous emotional mental state of a participant is in a particular one currently, he or she will most likely remain in that state. From table 1 it is clearly that the highest probabilities of each column are all distributed closely on the diagonal area, except for the surprise column (the data on diagonal is the second most likely for the surprise column). These facts indicate that full-blown emotions under neutral stimuli usually do not change in the short time anyway. They will retain the current emotional mental state if there is not any typical affective effecting.

Secondly, the Surprise mental state is a special emotional state in which the transition probability is absolutely different from the others six emotional state under neutral condition. In our experiment, the highest transition probability from the surprise state under neutral condition is to the calm state but not to the surprise state itself. According to the Circumplex Model of Affect [4], surprise is a high arousal and positive emotion. A person will not remain in a positive or highly intense state for a very long time if the stimulating effect is swift. After

**Table 1.** The Conditional Transitional Probability Tables

Under Neutral Conditions							
	happy	calm	sad	surprise	angry	fear	disgust
h	<b>0.444</b>	0.210	0.080	0.176	0.053	0.047	0.043
c	0.369	<b>0.547</b>	0.314	<b>0.287</b>	0.286	0.263	0.289
sa	0.054	0.082	<b>0.338</b>	0.082	0.118	0.121	0.085
su	0.051	0.048	0.052	0.257	0.069	0.094	0.051
a	0.025	0.036	0.093	0.086	<b>0.306</b>	0.086	0.147
f	0.028	0.040	0.059	0.068	0.060	<b>0.301</b>	0.068
d	0.029	0.038	0.064	0.044	0.109	0.088	<b>0.318</b>
Under Happy Conditions							
h	<b>0.597</b>	<b>0.575</b>	<b>0.463</b>	<b>0.523</b>	<b>0.433</b>	<b>0.401</b>	<b>0.463</b>
c	0.202	0.203	0.231	0.178	0.220	0.230	0.223
sa	0.028	0.029	0.098	0.043	0.046	0.044	0.035
su	0.134	0.151	0.135	0.201	0.155	0.153	0.131
a	0.013	0.014	0.030	0.020	0.093	0.026	0.034
f	0.016	0.017	0.022	0.017	0.023	0.115	0.031
d	0.011	0.011	0.023	0.019	0.030	0.031	0.082

a strong accidental stimulation, the emotion of a person will usually tend to transfer into the calm state and will not remain highly tense for a long time.

Thirdly, under neutral condition, the highest transition probability and the lowest one from one emotion state to another always corresponds to two states that are contradictory. According to the Cricumplex Model, for instance, happy which is a low arousal and positive value is absolutely opposite to angry, which has high arousal and negative value. From the data in table 1, it is easy to find several contradictory emotional state pairs. In our experiment, the contradictory emotion state pairs include: happy versus angry, calm versus angry, sad versus surprise, fear versus happy and happy versus disgust.

Fourthly, besides remaining in the same state, the probabilities for transferring into a neutral mental state are obviously higher than into the other emotional states. In table 1, we can find that the probabilities for transferring into the calm state are always on the first or second highest order. This shows that a full-blown emotion is a short time psychological phenomena and a person emotional state has a tendency to be calm if there are no outside stimuli continuously.

#### 4.4 Model Test

From the previous part a practical MST network model has been constructed that depends on about experiment with 200 questionnaires. To test the practicality of this MST network model, we used another set of 50 random survey



results as the test data. They are different from the former data and be employed for test task.

As the aim of the mental state transition modeling is used to predict the processing of emotion state transition from a previous state with its stationary transitional probability distribution and external condition. A person’s emotional action according to prediction of our transition network model will certify the validity of the model.

Firstly, we will verify the validity qualitatively. Comparison of the top two states transitioned from each state between the test data and corresponding model states probability distribution. The model can be proved to be useful when the states are matching and to be invalid when the states are not. Then, we test the model by comparing the transitional probability distribution of all the states. This will finally present a determinated probability that describes the level of the validity of the model.

**Table 2.** Qualitatively Test Result of Mental State Transition Modeling

Under neutral/ha/sa/su/a/f/d emotional condition							
	happy	calm	sad	surprise	angry	fear	disgust
happy	1	2					
calm	2	1	2	1	2	2	2
sad			1				
surprise				2			
angry					1		
fear						1	
disgust							1

In the first phase, the first two states with largest probabilities are selected to compare with the two from the test data directly, which are filled out by the participants. table 2 has shown one example result of transition in happy situation. The results for this part of the comparison have indicated that the model is valid qualitatively.

In the probability comparison case, the two kinds of transition probabilities,  $P_i(a_i|a_j)$  and  $Q_i(a_i|a_j)$  are considered.  $P_i(a_i|a_j)$  indicates the probability from state  $a$  to state  $b$  in the model and  $Q_i(a_i|a_j)$  is the probability calculated from the test data. The probabilities are the foundation of our comparison.

$$\sum_i P_i(a_i|a_j) = 1 \tag{1}$$

$$\sum_i Q_i(a_i|a_j) = 1 \tag{2}$$

In our model and test data, there are seven possible states to be transitioned into from the start state. In an ideal case, the distribution of the transitional

probability of the test data must match the model. We use the difference between the model and the test data to evaluate the validity. The following equation is used to calculate the related difference of the states. The equation describes the difference of one start state between the probability distributions,  $P_i(a_i|a_j)$  and  $Q_i(a_i|a_j)$ . As the difference increases, the result decreases. If the distributions of the probability are analogous, the result becomes one. For the whole model, we use the mean value of all states to evaluate the model validity. The equation is as follows:

$$P_r = \frac{1}{N} \sum_j \sum_i P(a_i|a_j)(1 - |P(a_i|a_j) - Q(a_i|a_j)|) \tag{3}$$

$N$  is the total number of all states.  $P_r$  ranges from 0 to 1. The closer to 1 the more valid the model is.

Compared with the 50 random test data, the probability of the model validity distributes on seven various external situations are indicated in the table 3.

It means the model is close to the actual emotional mental state transition model.

**Table 3.** The Conditional Transitional Probability Tables

The Model Validity Distributes on Various Situations							
	neutral	happy	sad	surprise	angry	fear	disgust
P	0.87	0.87	0.85	0.82	0.84	0.85	0.83

## 5 Virtual Character Application

From experimental results we can find that upcoming mental state depends on previous mental state and emotional event. In here, we build a framework of virtual character of affective interaction which implement the MST.



**Fig. 4.** Interface of framework of virtual character

The fundamental component of a virtual character is that its mental activity basically according to mental state transition modeling. In our work, MST itself can describe the human mental state transition and situation credibly and relatively accurate. The interface of the framework of virtual character is like in Fig. 4.

The framework depends on the dialog between user and the virtual character. First to analyze the impact from the input sentence from user, the response of the virtual character is expressed through its facial expression which is predicted by the MSTN indeed.

## 6 Conclusion and Future Work

The human's mental states are independent and thus we model a framework of mental state analysis including recognizing and synthesizing processing and mental state transition which are fundamental of constructing describe a human affective mind and making up a virtual character's emotional states. To model the MST, we also did a experiment and analysis of MST modeling. In the future, we plan to improve the affective virtual character interface and demonstrate the feasibility of the proposed framework. The whole architecture will also be combined with the affective-detection system and affective synthesis system into a human computer interaction platform.

## References

1. Rosalind, W.: *Picard: Affective Computing*. The MIT Press Cambridge, Massachusetts London, England (1997)
2. Ekman, P.: *Universals and Cultural differences in Facial Expressions of Emotion*. In: Cole, J. (ed.) *Nebraska Symposium on Motivation*, vol. 19, pp. 207–283. University of Nebraska Press, Lincoln (1972)
3. Plutchik, R.: *Emotions: A Psychoevolutionary Synthesis*. Harper, Row, New York (1980)
4. Russell, J.A.: *A Circumplex Model of Affect*. *Journal of Personality and Social Psychology* 39, 1161–1178 (1980)
5. Li, X., Li, Q.: *Active Affective State Detection and User Assistance With Dynamic Bayesian Networks*. *IEEE Trans. on System, Man and Cybernetics-Part A. Systems and Humans*, 93–105 (2005)
6. Cohen, I., Garg, A., Huang, T.S.: *Emotion Recognition from Facial Expression using Multilevel HMM*. *Neural Information Processing Systems* (2000)
7. Peilin, J., Hua, X., Fuji, R., Shingo, K.: *An Advanced Mental State Transition Network and Psychological Experiments*. In: Yang, L.T., Amamiya, M., Liu, Z., Guo, M., Rammig, F.J. (eds.) *EUC 2005. LNCS*, vol. 3824, pp. 1026–1035. Springer, Heidelberg (2005)
8. Tsuji, K., Okuda, T.: *Analyses of the Discomforts Aroused by Stimulus Sentences with Reference to Effects of Modality and Gender*. *The Japanese Journal of Research on Emotion* 3(2), 64–70 (1996)
9. Hua, X., Peilin, J., Shuang, X., Fuji, R., Shingo, K.: *A model of mental state transition network*. *IEEJ Transactions on Electronics, Information and Systems* 127(2), 434–442 (2007)

# Integration of Symmetry and Macro-operators in Planning

Amirali Houshmandan, Gholamreza Ghassem-Sani, and Hootan Nakhost

Sharif University of Technology,  
Department of Computer Engineering, Tehran, Iran  
Houshmandan@ce.sharif.edu, Sani@sharif.ir, Nokhost@ce.sharif.edu

**Abstract.** Macro-operators are sequences of actions that can guide a planner to achieve its goals faster by avoiding search for those sequences. However, using macro-operators will also increase the branching factor of choosing operators, and as a result making planning more complex and less efficient. On the other hand, the detection and exploitation of symmetric structures in planning problems can reduce the search space by directing the search process. In this paper, we present a new method for detecting symmetric objects through subgraph-isomorphism, and exploiting the extracted information in macro-operator selection. The method has been incorporated into HSP2, and tested on a collection of different planning domains.

## 1 Introduction

The detection and exploitation of symmetric structures in search problems such as AI planning can greatly reduce the size of the space that must be explored. Symmetry arises in different ways. Consider a robot that must transfer a set of balls from one location to another. This problem is simple for human who can observe the underlying symmetries. However, for a planner balls are distinguished by their names, and they are not regarded as similar objects and planner will plan for each ball independently from scratch. Moreover, the sequence of transferring these balls is not important. Taking all these facts into account, the search space would be reduced dramatically.

### 1.1 Symmetry in Planning

Different approaches have been proposed for identification of symmetric structures and improving the search process [1], [2], [3].

In [1], symmetry is considered as an automorphism of the problem definition, i.e. a function that maps the structure of a problem onto itself. A colored graph is constructed for each problem, which is based on predicates presented in the initial and goal states of the problem. The graph presents object relationships in problem definition. This graph is fed to NAUTY a graph automorphism discovery tool, for identification of automorphisms. Almost symmetric objects are spotted after restricting the NAUTY's output to domain objects. These objects are called almost

symmetric because they have some differences in their initial or goal configurations, though their behavior is essentially equivalent.

In [2], two phase static analyzing process is proposed for symmetry detection. They defined symmetric objects to be those which are indistinguishable from one another based on their initial and goal predicates. Their strategy was to begin with identifying pairs of symmetric objects as a base for symmetry groups, and then extending these groups by adding other objects of the same type. The second stage of their process was the identification of symmetric actions according to some particular parameters.

In [3], a different method for building the graph representing relations among predicates in the initial and goal states is used. This graph is then processed by NAUTY and generators for the graph's automorphism group are restricted to domain objects. These objects form symmetric groups.

## 1.2 Macro-operators in Planning

A macro-operator is a sequence of actions that can be treated as a single operator. Macro-operators can improve the performance of a planner because the planner does not need to plan about multiple steps encapsulated in the macro operator from the scratch. On the other hand, the addition of macro-operator increases the branching factor and can have negative effect on the planner's performance.

In [4], a new method for generating and filtering macro-operators was proposed. Their method uses a four step strategy. First, some structural information is extracted from the domain by analyzing. Then based on these structures a number of macro-operators are generated. Useful macro-operators are kept by applying a filtering and ranking procedure. Finally, these macro-operators are used in the planning process. This method has some limitations such as the size of macro-operators and dependency to the static predicates in definition of the domains. In [5], they have extended their method to overcome these limitations. In this work, macro-operators are generated based on problem's solutions. Partial ordering of primary operators is allowed in macro-operators. Their results show an impressive potential to reduce the search efforts when such macro-operators can be generated.

Macro-operators are also used to generate plateau-escaping macro actions in Marvin planner [6]. These macro actions are later used in the search process to solve new problems. Plateaux occur when a local minima in search space has been reached. It is often the case that the same sequence of actions with different parameters is used to escape these plateaux [7].

In this paper, we propose a new method for symmetry detection to be used as a guide for macro-operator's application in planning process. The main contribution is the integration of the two methods: symmetry and macro-operator. Our method has two general steps. It learns a set of macro-operators and their related graphs. Learning is based on some training problems for each domain. Each macro-operator is extracted from a training set, and then filtered according to a testing set. At the end of this stage, a reference model, a set of pairs consisting of a macro-operator and a graph, is generated for each domain. The second step is the usage of the reference model in new problems. A graph for input problem is constructed, and then compared to each entry in the reference model, by the subgraph isomorphism algorithms. If two

graphs are matched, the corresponding macro-operator will be extracted. These macro-operators are added to the domain definition for helping the planner to solve the input problems in a more informative way.

Our mechanism has been integrated into HSP2 [8], [9]. HSP2 is a heuristic search planner that uses weighted A\* search in the state space of the problem. In each search state, heuristic values are calculated from related relaxed problem in which delete lists are ignored. These information are kept by the planner and will be used during planning process. The proposed method is tested over a collection of different planning domains. The results show that the integration of symmetries and macro-operators can significantly improve the performance of the planner in different domains.

The structure of the paper is as follows. In section 2, the idea of primary predicates is introduced and a simple approach for identifying such predicates is explained. Then we discuss how to create a set of macro-operators in each domain. In section 3, we describe how to make the problem graph in detail together with the algorithm for exploiting symmetrical information in macro-operator's application. Finally, in section 4, we compare our results with the base planner and conclude the paper according to experimental results.

## 2 Identifying Primary Predicates and Learning Macro-operators

Identification of symmetric objects in previous approaches were mostly based on predicates appeared in the problem definition. In this work, the so-called primary predicate is the core of symmetry detection. Primary predicates can be inferred for a planning domain by simply analyzing a number of problems in that domain. The most frequent common predicate in the initial and goal states is taken as the primary predicate. In blocks-world domain, the primary predicate is  $(on\ ?x\ ?y)$ , in gripper domain it is  $(at\ ?x\ ?y)$ , and in the logistics domain it is  $(at\ ?x\ ?y)$ . The graph representation of the problem is constructed by these primary predicates. As a result, the size of graph remains small and computational cost for graph comparison will be reduced. In addition, as our experiments show, the extracted information by this mechanism would greatly reduce the search space in different domains.

After determining primary predicates, a set of useful macro-operators can be obtained. In this phase, two groups of problems are generated by a random problem generator. One set contains only simple problems, and form training set; the other is taken as the testing set. Problems in the training set are solved by HSP2. Their optimal solutions are candidates to form macro-operators. For each candidate, we have a testing phase. The graph of a simple problem is compared to the graphs of problems in the testing set through subgraph isomorphism algorithms. If matching occurred, the problem is solved by the use of this macro-operator in a way that will be discussed in the next section. Then, results of solving the problem with and without the macro-operator are compared. The candidate will be added to the reference model of the domain, in the case of improvement. Reference model is a set of pairs where each pair consists of a macro-operator and its corresponding graph. Pairs in the reference model are mutually exclusive. When two graphs in the model have intersection, the graph with fewer nodes is kept. Graphs also have relations between nodes and macro-operator's arguments.

At the end of this process, the reference model for each domain is obtained. In this approach there is no limitation on the number of arguments of macro-operator. The only condition is its improvement in testing phase.

### 3 Integration of Symmetry and Macro-operators

Although macro-operators can greatly improve the performance of the planning process, it has its own drawbacks. Increasing branching factor, as mentioned earlier, is one of the potential problems in macro-operator's usage [4], [5], [10]. In heuristic planners, the heuristic value is evaluated in each state. Thus the addition of macro-operators can increase the total number of these evaluations, because in each state all applicable macro-operators can be used. However, the application of most macro-operators in an active state may not be appropriate and leads to performance reduction.

Another issue in the state space planners is grounding of the operators. Presence of multiple arguments in macro-operators makes the grounding phase a time consuming task. In fact there is no need to ground these in all possible permutations; just a few of groundings are useful in each state. So restricting macro-operators applications to some predetermined domain objects can make them more efficient. The idea is discussed in this section through symmetry detection and subgraph isomorphism.

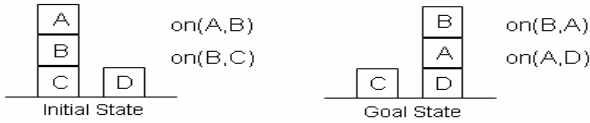
Previous approaches used symmetry either to choose similar actions from a collection of applicable actions for symmetric objects or to prune the search space in similar situations. Here, we propose a new approach to utilize subgraph isomorphism to restrict macro-operators in the planning process. These graphs are constructed based on primary predicates. The outline of the graph construction procedure is as follows:

1. Let  $G$  be a graph which is initially empty. The nodes and edges of  $G$  are colored as follows.
2. For each primary predicate in the Initial and Goal States do the following:
  - 2.1. For each argument  $a$ , if it has not been added yet, add a node with the default color to  $G$ . If  $a$  has a type, assign the related color to the node.
  - 2.2. Add a directed edge between predicate's arguments. If the predicate is in Initial State, the edge has color 1 and otherwise it has color 2.

After the creation of reference model and construction of problem's graph, the following steps will be performed:

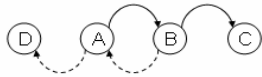
1. Select a graph from the reference model and compare it by the problem's graph:
  - 1.1. If they match, keep the related macro-operator from the reference model together with its mapping of variables.
  - 1.2. If they do not match, advance in the reference model and continue from 1.
2. If no matching is detected in step1, the algorithm will stop.
3. Add restrictive predicates to the preconditions of macro-operators.
4. Add macro-operators to the domain definition.
5. Add restrictive predicates to the problem definition.
6. Launch the planner with the updated domain and problem definitions.

Assume that one of the generated problems for macro-operator extraction is as follows:



**Fig. 1.** A simple example from blocks-world domain including Initial and Goal states with their primary predicates

In this simple example from the blocks-world domain, the goal is to reverse two blocks from one location to another one. Primary predicates for initial and goal states are also shown in picture. By applying the graph construction procedure on these predicates, we will get the following graph:

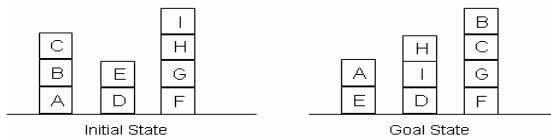


**Fig. 2.** Graph for our simple example constructed by related procedure. Filled lines show relations in initial state and dashed lines show relations in goal state.

Generated macro-operator for this problem comprises four primary operators:

MacroOp(A,B,C,D): Unstack(A,B), Stack(A,D), Unstack(B,C), Stack(B,A)

Now we consider the following planning problem in this domain. This problem is taken from HSP distribution package ( see Fig. 3) :

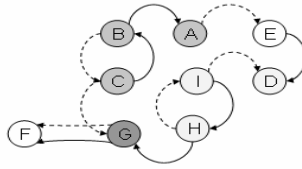


**Fig. 3.** Input problem that must be solved using macro-operators

By considering the primary predicates for this problem and constructing its graph, we can run the subgraph isomorphism algorithm to extract similar nodes. The result of running such an algorithm is depicted Figure 4.

In this problem, two groups of nodes are detected as being similar to the stored graph in the reference model. These groups are {C, B, A, G} and {I, H, G, D}. Now we can add the related macro-operator and restrict its usage based on these similar groups through some additional predicates. In our example, this predicate can be shown as (GL ?x ?y ?z ?m). "GL" predicates are added automatically by the system





**Fig. 4.** Graph for our input problem. Grayed nodes show matching groups. Node "G" matches in two separate groups.

after extraction of similar groups. These "GL"-prefixed predicates are generated for each entry in reference model that was matched with the problem's graph. Their arguments depend on the reference graph. In this case, four nodes are participating in the matching phase; thus the "GL" predicate consists of four arguments. This predicate is added to macro-operator's precondition, and then the specialized macro-operator is added to the domain definition. However as it was mentioned earlier, the usage of this macro-operator must be restricted to appropriate objects, i.e., members of similar groups. For capturing this, two predicates are added to the problem definition in this example. These predicates are (GL C B A G) and (GL I H G D). The ordering of arguments is determined by the reference model.

## 4 Experimental Results

In this section, we summarize our results from some experiments in five different planning domains and then compare the results of HSP2 with the result of our planner. Let us call out planner EHPS2, i.e., Enhanced-HSP2. We used VF2 [11], which is one of the fastest subgraph-isomorphism algorithms [12]. Some problems were taken from planning competitions AIPS-98, some were borrowed from HSP distribution package, and the rest were generated randomly using related random generators. For each problem, the number of generated nodes, expanded nodes, heuristic evaluations, and the overall CPU time with the time limit of five minutes, were considered.

The most time consuming steps in heuristic planners such as HSP2, is the heuristic evaluation. By using symmetry information, the application of macro-operators is restricted to those objects that are similar to the reference model. Thus, as mentioned earlier, if just a few grounded macro-operators are generated for each problem, not only the higher branching factor would not cause a problem, but also the search space would reduce considerably. By decreasing the generated states and exploiting the symmetry information, the count of heuristic evaluations is decreased.

The results for blocks-world domain are shown in tables 1 & 2. The blocks-world is challenging for domain-independent planners due to the interaction among subgoals and the size of state space. For this domain, five macro-operators formed the reference model. Three problems were taken from the HSP package and the rest were randomly generated. Other problems in the package were too small ( i.e., just contain small number of blocks) and, thus, were not included.

**Table 1.** Results for blocks-world domain without symmetry information

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob11.pddl <sup>1</sup>	721	201	721	.12
H_prob09.pddl	347	92	341	.05
H_prob06.pddl	37	21	37	.02

**Table 2.** Results for blocks-world domain using symmetry information

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob11.pddl	432	142	352	.08
H_prob09.pddl	22	6	18	.02
H_prob06.pddl	20	10	17	.01

Macro-operators are added in similar cases. Whenever some similarity is detected for an input problem, the addition of macro-operators will decrease the number of generated states. On the other hand when there is no such similarity, the problem will be solved in its usual way.

The results from the Gripper domain are shown in tables 3 & 4. The Gripper domain concerns a robot with two grippers that must transport a set of balls from one room to another. Solving these type of problems is simple for human, but the results of planning competitions showed that this domain is hard for some domain independent planners. For this domain, a single macro-operator is extracted and formed the reference model.

**Table 3.** Results for Gripper domain without symmetry information. G\_prob05 contains 12 balls and G\_prob06 contains 16 balls.

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob01.pddl	12	5	12	.01
H_prob02.pddl	31	9	31	.02
H_prob03.pddl	82	17	82	.03
H_prob04.pddl	155	25	155	.03
G_prob05.pddl <sup>2</sup>	338	37	<b>338</b>	.05
G_prob06.pddl	740	61	<b>740</b>	.08

In Gripper domain symmetry information avoids usage of macro-operators for balls that are not required to move from one room to another. Without this information, the macro-operator may be chosen for moving such balls and will generate many more states.

<sup>1</sup> Problems prefixed by "H\_" were taken from the HSP distribution package.

<sup>2</sup> Problems prefixed by "G\_" were randomly generated.

**Table 4.** Results for Gripper domain using symmetry information

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob01.pddl	18	4	8	.01
H_prob02.pddl	44	9	20	.02
H_prob03.pddl	129	24	46	.03
H_prob04.pddl	270	45	79	.03
G_prob05.pddl	660	152	<b>94</b>	.04
G_prob06.pddl	889	161	<b>205</b>	.05

Although in this domain the number of generated nodes is increased, the presence of a high degree of symmetry makes it possible to use heuristic values in similar states. Therefore, the number of heuristic evaluations is considerably decreased. In this domain the only learned macro-operator consists of three primitive operators: Pick, Move, and Drop. The algorithm makes this macro-operator applicable for each ball that needs to be moved. In fact, partial solution in the form of macro-operator dramatically improves the planner's performance. To make its impact more clear, consider the HSP2 results with  $W=1$  in Table 5. The results for  $W=1$  with symmetry information is as before.

**Table 5.** Results for Gripper domain without symmetry information and  $W=1$ 

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob01.pddl	12	5	12	.01
H_prob02.pddl	51	29	49	.02
H_prob03.pddl	762	286	678	.06
H_prob04.pddl	6430	2104	5750	.4
G_prob05.pddl	26000+	-	-	-
G_prob06.pddl	26000+	-	-	-

Results for logistics domain are shown in tables 6 & 7. Logistics domain involves the transportation of packages by either trucks or airplanes. There are several cities, each containing several locations, some of which are airports. Trucks drive within a single city, and airplanes can fly between airports. The goal is to get some packages from various locations to various new locations. For this domain, four macro-operators are extracted: Macros for transporting a package from city to city, from airport to city and vice-versa, and for transporting between two airports. The type information for domain objects and its usage in the graph's matching algorithm is greatly increasing the performance of the planner in this domain. As an example, in C\_prob03, the number of heuristic evaluations is decreased by about five times.

Tables 8 & 9 show the results in the Ferry domain. The problem in this domain is to transfer cars from a set of locations to another using a ferry. All locations are accessible from one another. This domain is similar to the gripper domain and the

symmetry information extracted from each problem has a notable impact on the search reduction. Only one macro-operator is extracted for this domain, which is similar to that of the gripper.

**Table 6.** Results for Logistics domain without symmetry information

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob01.pddl	10	3	10	.02
H_prob02.pddl	20	6	20	.03
H_prob03.pddl	30	9	30	.03
H_prob04.pddl	120	11	<b>120</b>	.09
H_prob05.pddl	236	20	<b>236</b>	.05
H_prob06.pddl	842	75	<b>842</b>	.27
C_prob01.pddl <sup>3</sup>	859	52	859	.22
C_prob02.pddl	4771	158	<b>4771</b>	2.62
C_prob03.pddl	25479	468	<b>25479</b>	84.49

**Table 7.** Results for Logistics domain using symmetry information

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob01.pddl	6	1	6	.01
H_prob02.pddl	6	1	6	.01
H_prob03.pddl	8	2	8	.02
H_prob04.pddl	17	1	<b>17</b>	.02
H_prob05.pddl	45	4	<b>45</b>	.02
H_prob06.pddl	348	26	<b>348</b>	.09
C_prob01.pddl	777	49	720	.21
C_prob02.pddl	399	8	<b>371</b>	.19
C_prob03.pddl	5787	100	<b>5545</b>	10.09

**Table 8.** Results for Ferry domain without symmetry information

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob01.pddl	9	8	9	.02
G_prob01.pddl	137	32	<b>137</b>	.03
G_prob02.pddl	401	117	<b>401</b>	.05
G_prob03.pddl	961	120	<b>961</b>	.07

Tables 10 & 11 show the result for Rockets domain. In this domain, objects must be transported between locations using rockets. Except three normal operators Load, Unload, and Move, there is an additional operator named Refuel in this domain.

<sup>3</sup> Problems prefixed by "C\_" were taken from planning competitions.

**Table 9.** Results for Ferry domain using symmetry information

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob01.pddl	9	4	6	.02
G_prob01.pddl	63	12	<b>18</b>	.03
G_prob02.pddl	368	44	<b>90</b>	.03
G_prob03.pddl	961	120	<b>178</b>	.05

**Table 10.** Results for Rockets domain without symmetry information

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob1.pddl	650	72	<b>637</b>	.11
H_prob2.pddl	359	43	359	.04

**Table 11.** Results for Rockets domain using symmetry information

Problem Name	Generated Nodes	Expanded Nodes	Heuristic Evaluations	Time
H_prob1.pddl	897	120	<b>265</b>	.06
H_prob2.pddl	387	46	263	.04

Rockets need fuel for moving between locations and after execution, the resource will be deleted. The reference model for this domain contains just one macro-operator.

## 5 Conclusion

In this paper, we proposed a new method to exploit symmetry information for optimizing the usage of macro-operators. The new method uses the notion of primary predicates to construct a structural graph from a given problem. Then, symmetric structures are extracted by applying subgraph isomorphism algorithms to the graph. It is shown that using primary predicates not only simplifies the process of extracting symmetries but also reveals very useful information regarding similar structures. Furthermore, analyzing the smaller resulted graphs is much faster. Our approach has been incorporated into HSP2 planner and tested on five different planning domains. The results showed the superior performance of our method comparing to HSP.

## References

1. Porteous, J., Long, D., Fox, M.: The Identification and Exploitation of Almost Symmetries in Planning Problems. In: Proceedings of the 23rd UK Planning and Scheduling SIG
2. Fox, M., Long, D.: The Detection and Exploitation of Symmetry in Planning Problems. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pp. 956–961 (1999)

3. Joslin, D., Roy, A.: Exploiting symmetry in lifted CSPs. In: AAAI, pp. 197–202 (1997)
4. Botea, A., Enzenberger, M., Mueller, M., Schaeffer, J.: Macro-FF: Improving AI Planning with Automatically Learned Macro-Operators. *Journal of Artificial Intelligence Research* 24, 581–621 (2005)
5. Botea, A., Müller, M., Schaeffer, J.: Learning Partial-Order Macros from Solutions. In: *Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling*, pp. 231–240 (2005)
6. Coles, A., Smith, A.: Marvin: Macro Actions from Reduced Versions of the Instance. Working Paper. Department of Computer and Information Sciences, University of Strathclyde, Glasgow, Scotland
7. Smith, A.: Extending the Use of Plateau-Escaping Macro-Actions in Planning. In: *International Conference on Automated Planning & Scheduling*, Cumbria, UK (2006)
8. Bonet, B., Geffner, H.: Planning as Heuristic Search. *Artificial Intelligence*, Special issue on Heuristic Search 129 (2001)
9. Bonet, B., Geffner, H.: HSP2. Description of HSP Planner in AIPS-2000 Competition, *AI Magazine* 22, 77–80 (2001)
10. García Durán, R.: Integrating Macro-Operators and Control-Rules Learning. In: *The International Conference on Automated Planning and Scheduling*, Cumbria, UK (2006)
11. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs. *IEEE* 26, 1367–1372 (2004)
12. Foggia, P., Sansone, C., Vento, M.: A Performance Comparison of Five Algorithms for Graph Isomorphism. *Graph-based Representations in Pattern Recognition* (2001)

# Planning by Guided Hill-Climbing

Seyed Ali Akramifar and Gholamreza Ghassem-Sani

Computer Engineering Department, Sharif University of Technology,  
P.O. Box 11365-9517, Tehran, Iran  
akrami@ce.sharif.edu, sani@sharif.edu

**Abstract.** This paper describes a novel approach will be called guided hill climbing to improve the efficiency of hill climbing in the planning domains. Unlike simple hill climbing, which evaluates the successor states without any particular order, guided hill climbing evaluates states according to an order recommended by an auxiliary guiding heuristic function. Guiding heuristic function is a self-adaptive and cost effective function based on the main heuristic function of hill climbing. To improve the performance of the method in various domains, we defined several heuristic functions and created a mechanism to choose appropriate functions for each particular domain. We applied the guiding method to the enforced hill climbing, which has been used by the Fast Forward planning system (FF). The results show a significant improvement in the efficiency of FF in a number of domains.

## 1 Introduction

Hill climbing is a local search technique, which has been widely used in artificial intelligence fields such as AI planning [1], machine learning [2], and optimization [3, 4]. It attempts to minimize (or maximize) a function  $h(s)$ , where  $s$  are discrete states. These states are typically represented by vertices (or nodes) in a graph, where edges (or actions) in the graph indicate nearness or similarity of vertices. Hill climbing traces the graph vertex by vertex, locally decreasing (or increasing) the value of  $h$ , until a local minimum (or maximum)  $s_m$  is reached.

Two major approaches of hill climbing are the so-called simple hill climbing and steepest ascent hill climbing. In simple hill climbing, the first closer node to the solution is chosen, whereas in the steepest ascent hill climbing; all successors are compared and the closest node to the solution is chosen. Simple hill climbing is more efficient than the steepest ascent hill climbing specially in domains with a high branching factor using a costly evaluation function. Therefore, it is preferred to use simple hill climbing in the planning domains.

Many research efforts have paid attention to improve efficiency of hill climbing. Examples include dynamic hill climbing, which uses a genetic algorithm technique to hill climbing [5], stochastic hill climbing, which uses a random order to evaluate successor states [6, 7], and hill climbing augmented with learning techniques [8].

The usual approach to overcome the efficiency problem is looking for a more efficient evaluation function [9]. To reach to the goal state faster, the evaluation

function needs to be more cost effective and more accurate. However, since we need to design a new evaluation function for each new domain, this is not a generally useful solution. Using more efficient domain independent heuristic functions is another important solution to reduce the efficiency problem [4]. Although, the domain independent option is more attractive than the previous one, it has its own limitations. First, because of difference in the nature of domains, usually the performance of a domain independent heuristic function varies in different domains. In other words, a domain independent heuristic function cannot significantly improve the efficiency of hill climbing in some domains. In addition, there is a tradeoff between the efficiency and the accuracy of a heuristic function. Usually, informed heuristic functions need more computational resorts and consequently are less efficient. On the other hand, uninformed but efficient heuristic functions evaluate states inappropriately. Therefore, this tradeoff reduces the total efficiency of the search.

We introduced a new approach called *guided hill climbing* (GHC), to improve the efficiency of hill climbing. In GHC, an auxiliary cost effective heuristic function  $g$  is used to order successor states. As simple hill climbing, GHC uses a primary heuristic function  $h$  to evaluate successor states according to the order proposed by  $g$ .

We applied GHC to the enforced hill climbing (EHC), which is the main search strategy of the Fast Forward (FF) planning system [10]. EHC is a combination of a systematic search method and simple hill climbing. We tested the efficiency and quality of new planner in a number of domains. Also, to improve the performance of GHC in various domains, we defined several heuristic functions and implemented a mechanism to automatically learn appropriate function  $g$  in each domain.

The remainder of this paper is organized as follows. Section 2 details the new approach to hill climbing. Section 3 explains our new planner, which employs GHC. We implemented this planner through applying GHC to FF. Section 4 presents results of several experiments demonstrating the advantages of GHC. This section also compares results of our planner with that of FF. Last section concludes.

## 2 Guided Hill Climbing

GHC is a different approach to overcome the efficiency problem of the simple hill climbing. Its search strategy is very similar to that of simple hill climbing in some ways. First, GHC employs an evaluation function to evaluate states. Furthermore, it searches to find a better successor state. Finally, GHC has basic problems of simple hill climbing such as facing a local maximum. However, it is different from simple hill climbing in that it uses an extra heuristic function called guiding heuristic function. By the means of this function, GHC orders successor states before evaluation by the main heuristic function of hill climbing. GHC algorithm is as follows:

```

Guided Hill-Climbing (state  $s_{in}$ ): state  $s_{out}$ 
  Let Open =  $\{s_1, \dots, s_k\}$  be successor states of  $s_{in}$ 
  While (Open is not empty) do
    Let  $s_i = g(\{s_1, \dots, s_k\})$ 
    if ( $h(s_i) < h(s_c)$ ) return  $s_i$ 
    else remove  $s_i$  from Open
    add successor states of  $s_i$  to Open
  return NULL

```



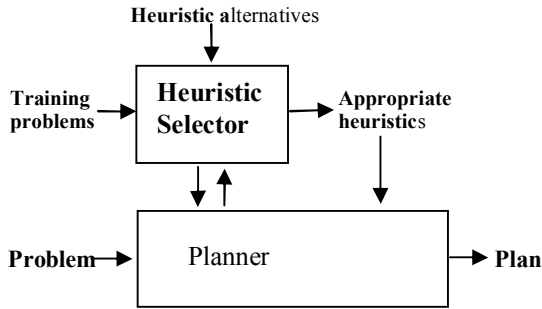


Fig. 1. The architecture of guided hill climbing planner (GHCP)

Suppose  $g$  is the function that can find a potentially appropriate successor state  $s_i$  for state  $s_c$ . GHC evaluates  $s_i$  by  $h$ . In the rest of this paper minimizing the  $h$  is assumed. If  $h$  reports  $s_i$  as appropriate (i.e., formally  $h(s_i) < h(s_c)$ ), GHC will replace  $s_c$  by  $s_i$ . GHC repeats until reaching to a goal state.

To improve the efficiency of guiding heuristic function,  $g$  needs to be a very low cost and accurate heuristic function. There is a tradeoff between efficiency and accuracy. To create an efficient hill climber, we have to resolve this tradeoff.

**Low Cost.** To avoid additional computational overheads, guiding heuristic function needs to be a very low cost function. Therefore, it must be as simple as possible. An appropriate solution is using a simple learning technique. Learning not only leads to a low cost heuristic function but also provides a more accurate one.

**More Accuracy.** On the other hand, guiding heuristic function must be as accurate as possible. Accuracy of guiding heuristic function depends on the main heuristic function. Therefore, the more accurate  $g$  is used the more appropriate state  $h$  evaluates. In other words, an accurate function usually recommends states that are more promising successor states.

GHC uses a statistical criterion, which we call *relative accuracy* of  $g$ , to determine the accuracy of  $g$ . We define the average relative accuracy as  $k_g = 100 * P/N$ , where  $P$  is the number of appropriate states that have been accepted by  $h$ , and  $N$  is the number of states that have been suggested by  $g$ . According to this definition, if  $k_g$  equals to 100,  $g$  will have the maximum average relative accuracy. Hence, a guiding function that has more average relative accuracy will reduce greater efforts of  $h$ .

### 3 Guided Hill Climbing Planner

Guided hill climbing planner (GHCP) is a new planner that incorporates the idea of guiding into the fast forward planner. Figure 1 shows the architecture of GHCP. GHCP has two main components: heuristic selector and planner. In each planning domain, heuristic selector determines an appropriate guiding heuristic function based on planning results of several experimental problems. Then, planner uses determined guiding heuristic function to plan new unseen problems.

There are two reasons in using FF as the base planner. First, FF uses EHC, which is a combination of systematic search and the simple hill climbing. This combination resolves common hill climbing problems (i.e., local maxima). Therefore, we can easily apply guiding heuristic function to this planner. Second, FF and its descendant planners such as Metric-FF [11], Conformant-FF [12], Contingent-FF [13], and Macro-FF [14] are among recent highly efficient planners. Therefore, if GHC improves the efficiency of FF, it can also improve the efficiency of other planners.

We designed the heuristic selector to evaluate different heuristics in different domains. For this purpose, we defined several alternative heuristics. For a particular domain, the heuristic selector examines the planning results of several experimental problems, and recommends appropriate guiding heuristic function for that domain.

Heuristic selector of GHCP selects appropriate heuristic in two steps. First, it selects three candidates for guiding heuristic function based on their average relative accuracy  $k_g$ . Then, it determines two different heuristics as the efficiency heuristic, and the quality heuristic. The efficiency heuristic is a recommended guiding heuristic to improve efficiency of GHCP, and the quality heuristic is for improving the quality of GHCP output plans.

### 3.1 Heuristic Function of FF

Two important aspects of a heuristic search planner are (a) a mechanism for goal distance estimation, and (b) a search strategy. FF estimates the goal distance through counting the number of actions in a relaxed plan, which is produced by ignoring some of existing restrictions of the planning operators [4]. It is often the case that the cost of an exact solution to a relaxed problem is a good heuristic estimation of the cost of solving the original problem. With regard to the search strategy, FF uses the so-called EHC, which is an extension of the ordinary simple hill climbing, combined with a local and a systematic search. Helpful actions pruning relieves the planner from looking at too many superfluous successors of each world state. This saves time proportional to the length of the path that the planner goes through. Therefore, helpful actions prune unnecessary successors of each state during a breadth first search, i.e., they cut down the branching factor. However, helpful actions pruning does not preserve the completeness of the search. Hence, if the EHC fails to solve a problem, FF retries to plan by a best first search strategy.

### 3.2 Alternative Guiding Heuristic Functions

In this section, we introduce eight different alternatives of guiding heuristic functions. In a particular domain, GHCP selects an appropriate alternative based on some results from several simple problems. We define these typical alternatives to evaluate the performance of GHCP. However, it is clear that one may as well similarly define some other alternative heuristic functions.

**G<sub>1</sub>: Minimum Failure Heuristic.** G<sub>1</sub> suggests a successor state that relevant action of which has had fewer failures. Relevant action for a state  $s_a$ , is an action  $a$  which has produced  $s_a$  from its predecessor state  $s_c$ . With the word failure, we mean the number of unsuccessful evaluations of previously states  $s_a$  by  $h$  (i.e., formally  $fail(a) = \text{number of states } s_a, \text{ where } h(s_a) > h(s_c)$ ), during planning for a particular problem. The initial value of the failure counter is zero.

**G<sub>2</sub>: Maximum Failure Heuristic.** Opposite to G<sub>1</sub>, G<sub>2</sub> selects a successor state that its relevant action has the highest failure record.

**G<sub>3</sub>: Minimum Distance Failure Heuristic.** Similar to G<sub>1</sub>, G<sub>3</sub> uses failure of actions, but it uses failure distance rather than the number of failures. Here, when  $h(s_i)$  fails, GHCP adds  $h(s_i)-h(s_c)$  to the failure distance of corresponding action  $a_i$ . In any state, G<sub>3</sub> selects those actions that have minimum failure distances.

**G<sub>4</sub>: Maximum Relative Failure Heuristic.** Opposite to G<sub>3</sub>, G<sub>4</sub> prefers those actions that have maximum failure distances.

**G<sub>5</sub>: Mostly appeared in Relaxed Plans.** In G<sub>5</sub>, those actions that belong to some relaxed plans are more important than others. Therefore, G<sub>5</sub> recommends a state  $s_i$  that its relevant action has been included in more previously produced relaxed plans.

**G<sub>6</sub>: Not G<sub>5</sub>.** Opposite to G<sub>5</sub>, G<sub>6</sub> recommends a state  $s_i$  that its relevant action has been included in fewer previously relaxed plans.

**G<sub>7</sub>: Mostly Appeared in Helpful Actions.** FF uses a concept, called helpful action, to reduce the number of potentially appropriate successor states. In G<sub>7</sub>, those actions that belong to helpful actions are important. Therefore, G<sub>7</sub> recommends a state  $s_i$  that its relevant action has been regarded as a helpful action more than others.

**G<sub>8</sub>: Not G<sub>7</sub>.** Opposite to G<sub>7</sub>, G<sub>8</sub> suggests a state  $s_i$  that its corresponding action has been regarded as a helpful action in fewer occasions.

During planning, GHCP will choose one of G<sub>1</sub> to G<sub>8</sub> by using its heuristic selector.

### 3.3 Heuristic Selector

A desirable planner is one that is efficient in many different domains. Various domains are different in many ways such as average branching factor, number of operators, dependency of operators, etc. Accordingly, GHCP has different level of efficiency in different domains.

To resolve this problem, GHCP comprises a strategy selection module, will be called "*heuristic selector*," to select more appropriate heuristic functions from a set of alternatives in each new domain. Here, the heuristic selector chooses the more appropriate heuristics from a set of eight predefined heuristic functions. After solving many random problems from a particular domain, heuristic selector decides which heuristic(s) is the most appropriate.

The heuristic selector module works as follows: It first solves many initial problems for a particular domain by GHCP, using all eight alternative heuristic functions. It then determines three high priorities heuristics, based on *average relative accuracy* of  $g$ . Finally, one of the candidate heuristics is selected as being the quality heuristic and one is selected as being the efficiency heuristic. The efficiency heuristic is a recommended guiding heuristic to improve the efficiency of GHCP, and the quality heuristic is for improving the quality of its output plans.

The appropriate heuristic selection phase determines final appropriate heuristics for a particular domain. If G <sub>$i$</sub>  is a selected heuristic, we will use G <sub>$i$</sub> HCP, instead of GHCP, to refer to the appropriate version of the planner in that domain.

## 4 Results and Discussion

In this section, we compare GHCP with FF in several classical planning domains. The FF planning system v2.3 [10] is a highly efficient heuristic search planner that has been implemented in C, won the 2<sup>nd</sup> International Planning Competition, and its posterior Metric-FF [11] was generally the top performer in the STRIPS and simple numeric tracks of the 3<sup>rd</sup> International Planning Competition.

Our benchmark domains included the classical Logistics and Blocks world of the 2<sup>nd</sup> IPC, and Rovers, and Satellite of the 3<sup>rd</sup> IPC.

Since EHC may stick to some dead-ends in some planning domains, we selected only deadlock free domains. In deadlock domains such as Mprime, and Depots, EHC cannot solve some problems. Therefore, FF is not a suitable planner for applying and evaluating our idea in these domains. As a result, evaluating impacts of applying the guidance idea in these domains needs another different deadlock free implementation of simple hill climbing.

Our main objectives of the experiments were follows: To evaluate the accuracy of the heuristic selection part in choosing appropriate strategies, to evaluate the guiding hill climbing efficiency in different planning domains, and to compare the efficiency of GHCP against that of FF.

### 4.1 Results of Heuristic Selector

We tested all eight alternative heuristics on experimental problems of all selected domains. As mentioned earlier, the heuristic selection part first determines three candidates and then selects the appropriate heuristics.

#### Step 1: Candidate Selection

Table 1 shows average relative accuracy of predefined heuristics. These results were obtained from 10 initial problems of each domain.

**Table 1.** Average relative accuracy: Results obtained from 10 experimental problem of each domain (step 1)

Domain	Results								Candidates		
	G <sub>1</sub>	G <sub>2</sub>	G <sub>3</sub>	G <sub>4</sub>	G <sub>5</sub>	G <sub>6</sub>	G <sub>7</sub>	G <sub>8</sub>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>
Rovers	53.6	27.5	54.3	27.6	39.0	47.3	35.6	60.2	G <sub>8</sub>	G <sub>3</sub>	G <sub>1</sub>
Logistics	53.1	8.5	53.6	9.9	22.1	13.5	5.7	26.6	G <sub>3</sub>	G <sub>1</sub>	G <sub>8</sub>
Satellite	62.4	35.9	62.7	36.1	66.2	35.3	41.9	50.5	G <sub>5</sub>	G <sub>3</sub>	G <sub>1</sub>
Blocks	59.8	52.7	58.5	52.7	50.6	56.7	56.5	5.9	G <sub>1</sub>	G <sub>3</sub>	G <sub>6</sub>

There are some points in result of step 1. First, in the logistics domain, the last candidate (i.e., G<sub>8</sub>) was significantly different from the two others. As table 2 shows, the heuristic selector rejected this heuristic in second selection step. Second, two heuristics G<sub>1</sub> and G<sub>3</sub>, were candidates in all domains. Therefore, these two heuristics are more general than others. Finally, G<sub>2</sub>, G<sub>4</sub> and G<sub>7</sub> were not a candidate in any domain. Therefore, it seems that these alternatives are not plausible in other domains.

**Step 2: Appropriate heuristic Selection**

Having completed the candidate selection phase, the process of heuristic selection begins. In this process, based on the average plan lengths and average number of nodes generated by the three heuristic candidates, the best quality and the most effective heuristic from these candidates are determined. In some domains, unique heuristic may be selected as both the best quality and the best effective heuristics.

Table 2 shows the result of the appropriate heuristic selection. In this step, heuristic selector selected appropriate heuristics (i.e., quality, and efficiency heuristics).

**Table 2.** Results of heuristic selection (step 2). Candidates obtained form step 1.

Domain	Average Plan Lengths				Average Nodes Generated			
	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	<i>h</i> -Quality	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	<i>h</i> -Efficiency
Rovers	23.7	23.9	23.3	G <sub>3</sub>	30.6	35.3	35.8	G <sub>8</sub>
Logistics	182.7	181.7	262.5	G <sub>1</sub>	325.3	330.4	4555	G <sub>3</sub>
Satellite	20.6	20.7	20.7	G <sub>5</sub>	31.3	33.0	33.1	G <sub>5</sub>
Blocks	21.4	21.7	21.4	G <sub>1</sub>	89.9	82.1	149.9	G <sub>3</sub>

There are some important points in this table. First, in the Satellite domain, both quality and efficiency heuristics are the same. Second, as mentioned before, in the logistics domain, the results of using G<sub>8</sub> are much worse than that of the two others. All selected strategies for the efficiency are the same as the first priority in the candidate selection phase, except for the blocks world, in which the second candidate was selected. As table 1 shows, the average relative accuracy of G<sub>1</sub> and G<sub>3</sub> are nearly the same (i.e., 59.8 ≈ 58.5). Therefore, the average relative accuracy seems to be an appropriate efficiency criterion.

Here, the heuristic selection process would finally propose the following heuristics (or planners) that should be employed in each domain: G<sub>1</sub>HCP and G<sub>8</sub>HCP for the Rovers domain, G<sub>1</sub>HCP and G<sub>3</sub>HCP for the logistics domain, G<sub>5</sub>HCP, for the satellite domain, and G<sub>1</sub>HCP and G<sub>3</sub>HCP for the blocks world.

**4.2 Results of Planning**

We tested GHCP by the means of selected guiding heuristic functions on several large problems, resulted as follows.

**Rovers Domain**

The Rovers domain inspired by the planetary rovers includes a collection of rovers that navigate a planet surface, analyzing samples of soil and rocks, and take images of the planet. Analyzing results and taken images are then communicated back to a lander. Parallel communications between rovers and the lander is impossible.

We tested FF, G<sub>1</sub>HCP and G<sub>8</sub>HCP on 20 different size problems. Problems of the Rovers domain were taken from the 3rd IPC. Table 3 shows the planning results. The table includes the lengths of plans (i.e., the number of generated actions) and the number of generated nodes. Data of each row belongs to one problem. The last row of the table shows the average performance of these planners. None of these planners solved problem 10 in a reasonable time (10 minutes). Therefore, the results based on 19 solved problems.

**Table 3.** Results for Rovers domain. Plan lengths and nodes generated for 19 problems.

Prob.	Plan Lengths			Nodes Generated		
	FF	G <sub>1</sub>	G <sub>8</sub>	FF	G <sub>1</sub>	G <sub>8</sub>
1	10	10	12	14	12	13
2	8	8	8	10	10	9
3	13	13	14	20	16	16
4	22	24	23	53	32	27
5	38	39	38	189	61	54
6	18	18	20	37	23	24
7	28	28	28	96	51	36
8	33	33	34	125	56	55
9	37	37	36	199	61	41
10	-	-	-	-	-	-
11	37	37	37	92	74	52
12	19	19	21	35	25	23
13	46	49	53	327	197	77
14	28	32	31	71	47	38
15	42	44	44	281	97	54
16	46	44	45	468	74	59
17	49	55	56	246	90	66
18	42	46	47	307	106	78
19	74	74	70	1144	159	95
20	96	97	94	2176	273	156
Average.	36.1	37.2	37.4	310	77.1	51.2

In average, all three planners produced plans with nearly the same lengths, but G<sub>1</sub>HCP and G<sub>8</sub>HCP were significantly more efficient than FF, especially in larger problems. In other words, the efficiency of G<sub>8</sub>HCP and G<sub>1</sub>HCP, are respectively 6.1 and 4.0 times higher than that of FF. Furthermore, G<sub>8</sub>HCP was more efficient than G<sub>1</sub>HCP and reversely G<sub>1</sub>HCP solutions had a better quality than that of G<sub>8</sub>HCP. These relations confirm the result of the heuristic selection part for Rovers, which suggested G<sub>8</sub>HCP for the efficiency and G<sub>1</sub>HCP for the quality heuristics.

### Logistics Domain

The Logistics is a classical domain involving the transportation of a number of packets by a number of trucks and airplanes. We tested G<sub>1</sub>HCP, G<sub>3</sub>HCP and FF on 20 different size problems. Problems of Logistics were taken from the 2<sup>nd</sup> IPC.

In the Logistics domain, the guidance idea had considerable improvement in the efficiency of hill climbing. As table 4 shows, FF could not solve the four largest problems (i.e., problems 17, 18, 19, and 20) in a reasonable time (10 minutes), while G<sub>1</sub>HCP and G<sub>3</sub>HCP solved all those problems. The last row of the table shows the average performance of these planners based on 16 solved problems by FF. G<sub>1</sub>HCP and G<sub>3</sub>HCP solved each problem more efficiently than FF. Consequently, the average number of nodes generated by FF was 17.5 times greater than that of G<sub>1</sub>HCP, and was 19.8 times greater than that of G<sub>3</sub>HCP. In fact, the improvement in the efficiency is even more significant than what is shown in these experimental results, because we have considered only those problems that FF was able to solve.

On the other hand, FF produced slightly shorter plans in some problems such as problems 4 and 5. That is because G<sub>1</sub>HCP and G<sub>3</sub>HCP, for the sake of improving the efficiency, prune more search spaces than FF, and as a result, may overlook some smaller plans produced by FF. The plans produced by the G<sub>1</sub>HCP and G<sub>3</sub>HCP are in

almost all the cases as good (small) as that of FF. In other words, having preserved the quality of the plans produced by FF, GHCP improved the speed of the planning in the Logistics domain.

**Table 4.** Results for Logistics domain- Average presented in the last row is calculated for only first 16 problems

Prob.	Plan Lengths			Nodes Generated		
	FF	G <sub>1</sub>	G <sub>3</sub>	FF	G <sub>1</sub>	G <sub>3</sub>
1	27	27	27	35	32	32
2	46	49	49	90	61	61
3	82	82	81	221	118	111
4	118	126	123	569	162	182
5	159	166	172	914	255	303
6	187	213	206	1003	364	314
8	251	275	246	2050	544	431
7	193	217	240	3492	436	539
12	395	420	398	9176	953	755
10	324	346	371	10938	750	683
13	398	464	453	12209	1312	1154
9	298	316	312	13006	582	597
14	451	498	508	22286	1298	1359
11	353	400	385	25671	868	1153
15	505	527	523	45785	1383	1181
16	535	677	592	52613	2294	1248
17	-	628	587	-	1774	1639
18	-	689	666	-	2254	2243
20	-	751	758	-	2255	2460
19	-	749	693	-	2304	1740
Average.	270.1	300.2	292.9	12503.6	713.3	631.4

**Satellite Domain**

The satellite domain is intended to be a first model of a satellite’s observation scheduling problem. The full problem involves using one or more satellites to make observations, collecting data, and down-linking the data to a ground station.

We tested G<sub>5</sub>HCP and FF on 20 different size problems. Problems of the Satellite domain were taken from the 3<sup>rd</sup> IPC. As Table 5 shows, in this domain, G<sub>5</sub>HCP was more efficient than FF, especially in larger problems (e.g., problems 19 and 20). In addition, G<sub>5</sub>HCP, in average had a better performance than FF. G<sub>3</sub>HCP improved both efficiency and quality of FF in this domain.

**Blocks World**

In the Blocks world, we tested G<sub>1</sub>HCP, G<sub>3</sub>HCP and FF on 20 different problems. Problems were borrowed from the 2<sup>nd</sup> IPC. Neither of these planners could solve four last problems in a reasonable time. Therefore, as reported in table 6, the results of only 16 problems are compared. This table shows the plan length and the number of nodes generated for each problem. The last row shows the average performance of these planners based on the solved problems.

In the Blocks world, the quality of G<sub>2</sub>HCP and G<sub>1</sub>HCP is slightly better than that of FF. Besides, the efficiency of G<sub>2</sub>HCP and G<sub>1</sub>HCP is considerably more than that of FF (G<sub>1</sub>HCP was 13.9 times, and G<sub>3</sub>HCP was 14.7 times more efficient than FF). Similar to the Logistics domain, efficiency improvement of guiding hill climbing in the Blocks world domain is noticeable.

**Table 5.** Results for Satellite domain

Prob.	Plan Lengths		Nodes Generated	
	FF	G <sub>5</sub>	FF	G <sub>5</sub>
1	9	9	15	15
2	13	13	24	24
3	11	11	19	19
4	18	18	27	19
5	16	16	28	28
6	20	20	47	42
7	22	22	54	39
8	28	28	54	43
9	35	34	73	43
10	35	35	87	39
11	34	34	91	56
12	43	43	91	82
13	61	61	243	140
14	42	42	84	63
15	52	49	182	128
16	53	53	180	137
17	48	50	152	74
18	35	35	75	50
19	73	70	365	200
20	107	105	5889	4790
Average.	37.8	37.4	389	301.6

**Table 6.** Results for the Blocks World domain. Plan lengths and node generated for 16 solved problems.

Prob.	Plan Lengths			Nodes Generated		
	FF	G <sub>1</sub>	G <sub>3</sub>	FF	G <sub>1</sub>	G <sub>3</sub>
4-0	6	6	6	9	9	9
5-0	12	12	12	24	23	23
6-0	20	18	18	101	48	48
7-0	20	20	20	26	26	26
8-0	18	18	18	25	25	25
9-0	30	30	30	146624	10225	9244
10-0	34	34	34	63	49	49
11-0	34	34	34	83	61	61
12-0	44	42	42	1743	281	248
13-0	42	42	42	75	57	57
14-0	40	40	40	99	74	74
19-0	62	62	62	120	91	91
20-0	60	60	60	97	86	86
21-0	78	78	78	139	103	103
22-0	78	78	78	3416	159	180
23-0	76	76	76	98	98	98
Average.	40.9	40.6	40.6	9546.4	713.4	651.4

## 5 Conclusion and Future Works

This paper introduced guided hill climbing as an approach to improve the efficiency of the simple hill climbing. Guiding hill climbing uses an auxiliary heuristic function to sort successor states for consideration. Then the main heuristic function evaluates states according to the suggested order. We implemented a guided hill climbing planner (GHCP) through applying GHC to the fast forward planner (FF). FF planning system was chosen to evaluate the idea in the area of planning. GHCP works in two main phases. In the training phase, through solving some simple problems of a



particular domain, it selects an appropriate guiding heuristic functions out of a collection of predefined alternative functions. It then uses a selected heuristic function to plan complex problems of that domain. We first trained GHCP by using a number of small problems of four classical planning domains, and then it was tested on some more complex benchmark problems. The results show a significant improvement over FF with respect to the efficiency of the base planner. We think that application of GHC to other hill climbing based planners could improve their efficiency, too.

We consider four different directions for extending this work. First, we are exploring other appropriate guiding heuristic functions to further improve the performance of GHCP. We also plan to implement an independent simple hill climbing based planner to evaluate GHC on domains in which FF is trapped in a deadlock. We also hope to be able to show that the idea of GHC is applicable to other areas of AI, too. Another possible direction for further research is to apply the idea to more powerful search methods such as  $A^*$ , and its different derivations, which are applicable in many different areas of AI.

## References

1. Ghallab, M., Nau, D., Traverso, P.: *Automated Planning Theory and Practice*. Morgan Kaufmann, San Francisco (2004)
2. Mitchell, T.: *Machine learning*. McGraw Hill Inc, New York (1997)
3. Rich, E., Knight, K. (eds.): *Artifice Intelligence*. McGraw-Hill, New York (1991)
4. Russell, S.J., Norvig, P.: *Artifice Intelligence: A Modern Approach*. PrenticeHall, Englewood Cliffs (1995)
5. Yuret, D., Maza, M.: *Dynamic Hillclimbing: Overcoming the Limitations of Optimization Techniques*. In: *The Second Turkish Symposium on Artificial Intelligence and Neural Networks*, pp. 208–212 (1993)
6. Juels, A., Watenberg, M.: *Stochastic Hill-Climbing as a Baseline Method for Evaluating Genetic Algorithms*. Tech-Rep, University of California at Berkeley (1994)
7. Rudlof, S., Koppen, M.: *Stochastic hill climbing with learning by vectors of normal distributions*. Nagoya, Japan (1996), [citeseer.ist.psu.edu/rudlof97stochastic.html](http://citeseer.ist.psu.edu/rudlof97stochastic.html)
8. David, P., Kuipers, B.: *Learning hill-climbing functions as a strategy for generating behaviors in mobile robots*. TR AI90-137, University of Texas at Austin (1990)
9. Korf, R.: *Heuristic evaluation functions in artificial intelligence search algorithms*. *Minds and Machines* 5(4), 489–498 (1995)
10. Hoffmann, J., Nebel, B.: *The FF planning system: Fast plan generation through heuristic search*. *Journal of Artificial Intelligence Research* 14, 253–302 (2001)
11. Hoffmann, J.: *The Metric-FF planning system: Translating ignoring delete lists to numeric state variables* 20, 291–341 (2003)
12. Brafman, R., Hoffmann, J.: *Conformant planning via heuristic forward search: A new approach*. In: *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS-2004)*, Whistler, Canada (2004)
13. Hoffmann, J., Brafman, R.: *Contingent planning via heuristic forward search with implicit belief states*. In: *Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS-2005)*, Monterey, CA, USA, pp. 71–80 (2005)
14. Botea, A., Enzenberger, M., Mueller, M., Schaeffer, J.: *Macro-FF: Improving AI Planning with Automatically Learned Macro-Operators* 24, 581–621 (2005)

# DiPro: An Algorithm for the Packing in Product Transportation Problems with Multiple Loading and Routing Variants\*

Laura Cruz Reyes, Diana M. Nieto-Yáñez, Nelson Rangel-Valdez,  
Juan A. Herrera Ortiz, J. González B., Guadalupe Castilla Valdez,  
and J. Francisco Delgado-Orta

Instituto Tecnológico de Ciudad Madero, México  
lcruzreyes@prodigy.net.mx, diananieto@gmail.com,  
{ia32, juanarturo\_, jjgonzalezbarbosa, jgvaldez50}@hotmail.com,  
francisco.delgado.orta@gmail.com

**Abstract.** The present paper approaches the loading distribution of trucks for Product Transportation as a rich problem. This is formulated with the classic Bin Packing Problem and five variants associated with a real case of study. A state of the art review reveals that related work deals with three variants at the most. Besides, they do not consider its relation with the vehicle routing problem. For the solution of this new rich problem a heuristic-deterministic algorithm was developed. It works together with a metaheuristic algorithm to assign routes and loads. The results of solving a set of real world instances show an average saving of three vehicles regarding their manual solution; this last needed 180 minutes in order to solve an instance and the actual methodology takes two minutes. On average, the demand was satisfied in 97.45%. As future work the use of a non deterministic algorithm is intended.

## 1 Introduction

A common problem in production companies is the optimal transportation of its products; because a good handling involves the reduction from 5% to 20% of the total cost of the product [1]. This problem has been thoroughly studied. However, in our knowledge, that has not been enough for practical purposes. Rich problem is an emergent concept to denote real word problems.

The most recent researches approach real situations of transport with a complexity of up to five variants of the well-known Vehicle Routing Problem (VRP) [2]. The software commercial tools involve eight VRP variants at the most [3]. However, they do not contemplate the load distribution inside the trucks.

Bin Packing Problem (BPP), has the objective to search for the best accommodation of items into bins. Several researches have developed solution algorithms with excellent results. However, they do not contemplate important constraints derived of the real operation; besides the work with VRP. Some works are

---

\* This research was supported in part by CONACYT and DGEST.

focused in the problematic of VRP taking as constraint the truck capacity, and only using BPP to determine the number of trucks [4]. In [5], VRP and two-dimensional BPP are approached with few constraints; leaving out others related with the fragility of items, diversity of trucks, highways with a limited vehicular weight, and trucks with loads from multiple clients.

This paper shows a methodology based on heuristics for the integral solution of a rich product transportation problem. The description is centered in the load of trucks by means of a deterministic algorithm. In order to validate the proposal, we use as a case of study, a company dedicated to the transportation of bottled products.

## 2 Routing, Scheduling and Loading Problem (RoSLoP)

Routing Scheduling and Loading Problem (RoSLoP), approaches three tasks: assignment of routes, assignment of schedules and assignment of the load. In RoSLoP three optimization objectives must be achieved: satisfy the demands of all clients, minimize the number of used vehicles and reduce the total time of the trip. The schedules and routes assignment is modeled using VRP, while the load assignment is formulated with BPP.

## 3 Bin Packing Problem (BPP)

Bin Packing is a classic problem of combinatorial optimization. BPP belongs to the NP-Hard class [6]. This class contains a set of problems considered intractable because they demand many resources for their solution. The quantity required by these is similar to an exponential function or a polynomial of high grade. The demonstration of its complexity is based on the partition problem reduction.

In BPP exists an unlimited number of bins with capacity is  $c$ , the number of items is  $n$ , the size of each item is  $s_i$  ( $0 \leq i < n$ ) and  $s_i$  is limited to  $0 < s_i \leq c$ . The goal is to determine the smallest number of bins  $m$  in which the items can be packed; to get a minimal partition of the sequence  $L$  of  $n$  items, where  $L = B_1 \cup B_2 \cup \dots, \cup B_m$  such that in each set  $B_j$  the sum of the size of each item in  $B_j$  does not exceed  $c$  (expression 1).

$$\min z = \sum_{s_i \in B_j} s_i \leq c \quad \forall j, 1 \leq j \leq m. \quad (1)$$

### 3.1 BPP Variants

The previous definition makes reference to the basic problem of one dimension (BPP 1D). However, the real application of this research involves additional variants. The following five variants were identified in [7]:

- BPP On-line (BPPOn). The total number of elements to be accommodated is not known at the beginning of the process. Not all previous knowledge is assumed.
- BPP Capacity Constrained (BPPCC). Each bin has a specific capacity [8].
- BPP Cardinality Constrained (BPPcC). This variant adds a limit in the number of items that can be placed in a bin [9].

- BPP with Fragile Objects (BPPFO). Each item has a threshold of maximum weight supported so that it does not suffer damage or deterioration [10].
- Multiple Destinations BPP (MDBPP). The items can be unloaded in multiples destinations. For this reason, the order of their accommodation is important [11].

### 3.2 Related Work

Table 1 summarizes some investigations around BPP with variants. It can be observed that in other works at the most three variants are involved. In contrast, our research tries simultaneously with five variants of BPP and six variants of VRP [14]. These conditions are needed to formulate real and complex problems.

**Table 1.** The State of the Art of BPP and their variants

Author \ Variants that it approaches	BPPCC	BPPcC	BPPOn	BPPFO	MDBPP
Epstein 2005 [9]	✓	✓	✓		
Chan 2005 [10]		✓	✓	✓	
Manyem 2002 [12]	✓		✓		
Correia 2006 [13]	✓				
Balogh 2005 [6]			✓		
Kang 2003 [8]	✓				
Verweij 1996 [11]					✓
This Work	✓	✓	✓	✓	✓

## 4 Description of the Case of Study

The complexity of RoSLoP is increased when the factors observed in real applications are taken in consideration. In the case of the studied company, the transportation of bottled products is subject to the following conditions:

- The existence of diverse depots with multiple schedules.
- Fleets of vehicles of diverse kinds and with different load possibilities, where the weight balance should be taken in account for their loading.
- Clients with a specific demand, a time of service, and certain capacity constraints according to quantity and unit kind.
- Goods to be distributed that are characterized by product boxes with different attributes, such as: size, weight, supported weight and kind.
- Routes that connect clients and depots with an associate cost of trip.
- Government constraints that limit the traffic of vehicles that can travel on some roads, or the weight of the goods that can be transported.

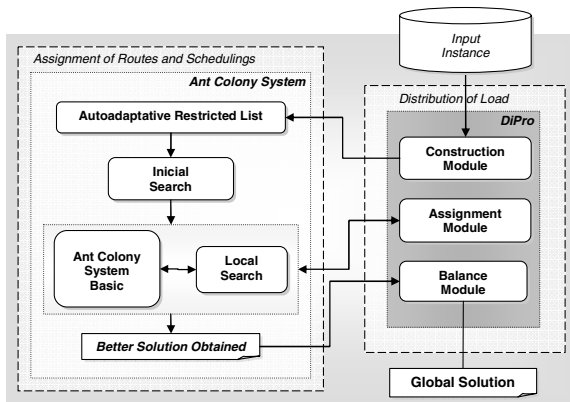
In the context of RoSLoP the objective of our rich BPP is to minimize the quantity of necessary vehicles to distribute the total load. For this study three particular objectives were derived: to maximize the number of assigned products, to maximize the utilized space of the vehicle and to balance the weight of the bins in each vehicle; according to the imposed conditions. In Table 2 are described the conditions of the case of study that have relationship with variants of BPP.

**Table 2.** BPP Variants of as part of the case of study

Variant	Operative Description
<i>BPPCC</i>	The bins in the vehicles can have different height and as consequence different loading capacity.
<i>BPPcC</i>	The maximum number of products that can be stacked in a bin are delimited by product kind, vehicle supported weight and the roads where it travels to.
<i>BPPOn</i>	The products to be distributed in a vehicle are not possibly known since the beginning. The assignment depends on the next client to be visited.
<i>BPPFO</i>	Each product has a quantity of supported weight that should not be overloaded to be able to avoid damage.
<i>MDBPP</i>	Since different discharge points exist in the same route, the product should be organized with base in this scheme.

## 5 Solution Methodology

The methodology proposed to solve RoSLoP includes an Ant Colony System (ACS) for solving VRP and the algorithm DiPro (Distribution Product) to solve BPP. Both of them coexist in an integral solution scheme of the tasks of routing, scheduling and loading (see Fig. 1). The ACS algorithm is a metaheuristic that uses closeness and pheromone measures to search routing solutions in a distributed way [14].



**Fig. 1.** Integral Solution Scheme of RoSLoP

The general algorithm begins invoking the Construction Module to transform items of three dimensions to one dimension. Then, an Autoadaptative Restricted List is created to determine the grouping level of the clients. The purpose of this list is to limit the set of candidate clients to be visited. Later, a greedy Initial Search is executed. Next, solutions are built through the basic ACS.

These solutions are improved by means of Local Search. During the construction of routes, every time that a new client is visited, the Assignment Module is invoked to distribute the load. To conclude the process, the Balance Module is invoked to get a balance of the best previous solution. Balancing is carried out considering the order of visit of each client and the weight of the product.

## 6 DiPro Algorithm

DiPro is a deterministic heuristic algorithm because it always obtains the same solution in different executions. This algorithm was designed to distribute and to assign the products in each one of the vehicles. The assignment is made with base in the clients that will be visited during the same route. DiPro is constituted by three modules applied in different times: construction, assignment and balance.

### 6.1 Construction Module

The purpose of the Construction Module is to convert items from three dimensions to one dimension. This process depends of the peculiar characteristics of each product. As example, the transformation is described with bottled products that are stored in units with special characteristic such as:

- **Boxes** (*Q*): Basic unit of an order from a client. With properties as height and weight; besides the supported weight that is a measure of the quantity of product that can be placed above a box without causing damage.
- **Beds** (*BEDS*): Constituted by a set of boxes, ordered in such a way that their length and width is adjusted at the bin of the truck. Some properties are: the number of boxes, depending of the product that composes them; and those inherent to the product, like height, weight, and supported weight.
- **Platforms** (*PLATFORMS*): Assignment unit that allows handing the products of three dimensions as a single dimension. Their properties are: kind (with values of homogeneous and incomplete, that later it will become heterogeneous), height, threshold of height, weight, supported weight (the smallest difference among the supported weight of each bed that conforms the platform, subtracted from the respective sum of the weight of the superior beds).

Fig. 2 shows the algorithmic outline of this module. The Construction Module carries out the order  $d$  of each client  $c$  to the different depots (lines 1 and 2), the translation of the information of the order that initially is in *Q* to *BEDS* and *PLATFORMS*. The function *homogeneous\_units()* (line 3) carries out the conversion of boxes to beds and later to platforms, by means of derivative calculations of the product properties; giving place to complete homogeneous platforms and remaining

homogeneous beds. Finally, in line 4 the function *incomplete\_platforms* () creates incomplete platforms with the rest of homogeneous beds; assuring that the beds of the same product remain in the same platform.

```

Construction Module (C, ORDERS)
1  for each c ∈ C
2      for each orderd ∈ ORDERSc
3          homogeneous_units (orderd)
4          incomplete_platforms (orderd)
5      end for
6  end for
    
```

Fig. 2. Algorithm of the Construction Module

### 6.2 Assignment Module

Their task consists on establishing the product accommodation inside the bins of the vehicle. This is executed while free space exists in the vehicle, the supported weight is not exceeded by this and the demand has not been satisfied in the order from the client to the depot. In each assignment the conditions of Table 2 are verified.

The Assignment Module uses the mechanism of Table 3 called completeness level (variable *completeness<sub>pallet<sub>x</sub> ∈ PALLET<sub>ζ, order<sub>d</sub></sub></sub>*) that allows verifying the restrictions imposed by BPPcC, BPPCC and BPPFO.

Table 3. Description of Completeness levels of an *order<sub>d</sub>* in a *pallet<sub>x</sub>*

Completeness Level	Description
0	Platforms have not been assigned
1	More homogeneous platforms cannot be assigned
2	More heterogeneous platforms cannot be assigned

Fig. 3 shows the Assignment Module Algorithm. This module is executed while the two conditions of line 1 are satisfied. The first one verifies by means the completeness level that free space exists in the trip  $\zeta$  of the vehicle  $v$  ( $v_\zeta$ ), and the supported weight is not exceeded by this. The second one verifies that the demand of the *order<sub>d</sub>* from the client *c* to the depot *d* has not been satisfied. In other words, some platforms exist without being assigned. In line 2 a bin of the vehicle  $v_\zeta$  is obtained; it is secured that the bin has space with a completeness level smaller than two. Line 3 allows the assignment of homogeneous and heterogeneous platforms only if: the *order<sub>d</sub>* has not satisfied the demand and that those space exists in the *bin<sub>x</sub>*.

Lines 4-9 locate the platforms, which give preference to the homogeneous in the function *assign\_homogeneous\_platform*() of line 4. In line 7, the method *assign\_incomplete\_platform* () assigns heterogeneous platforms that are formed from the incomplete. Both methods modify the completeness level when some of the constraints are violated.

---

**Assignment Module** ( $v_c, c, order_d \in ORDERS_c$ )

---

```

1  while  $\exists bin_x \in BINS_{v_c} \mid completeness < 2$ 
       $\wedge \exists platform_z \in PLATFORMS_{order_d} \mid isassigned = false$ 
2    Select  $bin_x \in BINS_{v_c} \mid completeness_{bin_x, order_d} < 2$ 
3    while  $\exists platform_z \in PLATFORMS_{order_d} \mid isassigned = false \wedge completeness_{bin_x, order_d} < 2$ 
4      if  $\exists platform_x \in PLATFORMS_{order_d} \mid kind = homogeneous$ 
           $\wedge completeness_{bin_x, order_d} = 0$ 
5        assign_homogeneous_platform ( $bin_x, order_d$ )
6      Else
7        assign_incomplete_platform ( $bin_x, order_d$ )
8      End if
9    end while
10 end while

```

---

Fig. 3. Algorithm of the Assignment Module

Fig. 4 summarizes the Assignment Module in the different scenarios that can be presented. The Spiral strategy is used to the assignment of homogenous and incomplete platforms. Initially all the elements are organized by descendent order of the supported weight. Then an element that satisfies all the conditions imposed by BPPcC, BPPCC, and BPPFO, is searched, alternating seeks in a descending and ascending way over the elements. This heuristic balances the weight of the goods sent in the different trucks. It is not allowed the programming of vehicles trips with very heavy or very light products.

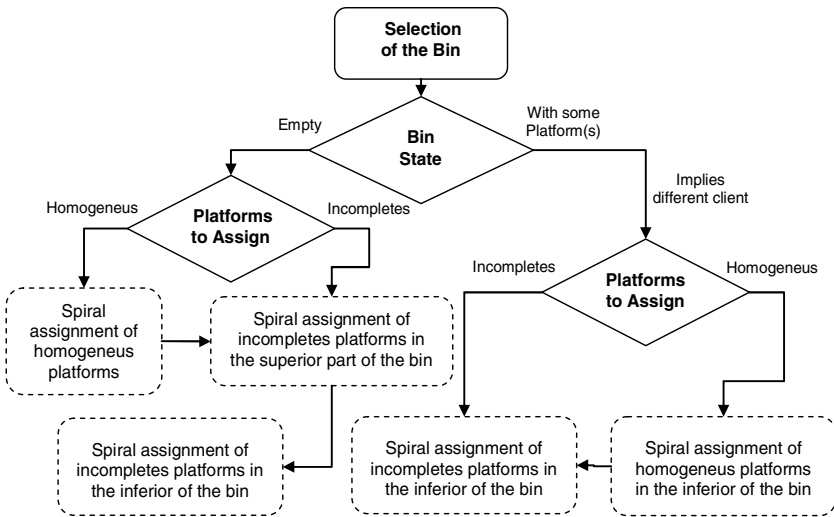
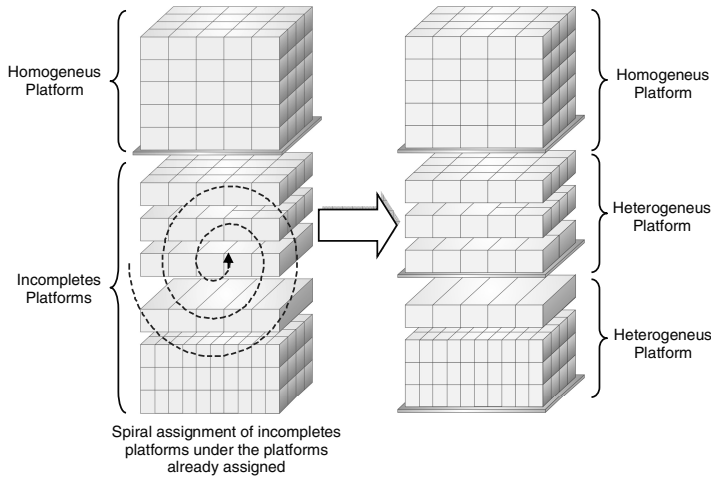


Fig. 4. Assignment Module



Fig. 5 shows the assignment of incomplete platform using the spiral strategy. As was mentioned, the homogeneous platforms are assigned first, because of its priority. Next, the incomplete platforms are converted in heterogeneous platforms using the spiral strategy. This conversion verifies all constraints derived from the possible formation of heterogeneous platforms. Finally, the converted platforms are placed under the homogeneous platforms.



**Fig. 5.** Conversion of incomplete to heterogeneous platforms using spiral assignment

### 6.3 Balance Module

In [14], Cruz approaches the requirement of balance regarding that the heaviest platforms should be located in the front part of the vehicle, while the lightest ones should go behind. Also, the total weight of the platforms that are in the right side of the vehicle should be approximate to the sum of those in the left side. However the specifications of the problem imply as a variant MDBPP; a balance was established which considers as first constraint, the sequence of visit to each client belonging to the same route. The product of the first client being visited will go in the back part of the unit and the last one in the front.

Fig. 6 details the Balance Module; which is of great importance in the process due to the problem with BPPOn, where the constraints established above are not possible to be verified in the assignment. As part of the method, all the platforms set assigned to the same bin are called product stack. Let  $N_v(v)$  the stack set from the vehicle  $v$  and  $BINS_v$  the set of all bins of  $v$ . Is it feasible to move a stack of products from one bin to another, if it fits in the bin of destiny according to its height. The bins are grouped initially. The class of each bin and each stack is determined in an ascendant function according to its height (lines 1 and 2). The product stacks are marked as not reassigned and the bins as not used (lines 3 and 4).

The product stacks are sorted in an ascendant way based on three aspects: class, stack weight and order of clients visit. In the last aspect, in case that the stack is

formed by products that belong to different clients, the reference is taken from the client to which belongs the upper platform of the stack (line 5).

If there exists a stack of product without being reassigned, an empty bin would be selected whose position would be nearest to the front of the vehicle. The biggest stack of product with the lowest class would be assigned into the pre-selected bin. The set  $BINS_v$  and  $N_\gamma(v)$  are upgraded, in lines 6-12.

---

**Balance Module** ( $v$ )

---

```

1  Assigning classes  $class_{bin}, \forall bin_x \in BINS_v$ 
2  Assigning classes  $class_\gamma, \forall \gamma \in N_\gamma(v)$ 
3  set  $\forall \gamma \in N_\gamma(v)$  : not reassigned
4  set  $\forall bin_x \in BINS_v$  : not used
5  Sort  $\gamma \in N_\gamma(v)$  by height/class, visit succession and weight
6  while  $N_\gamma(v) \neq \emptyset$ 
7      Select  $bin_x$  nearest in the front does not used
8       $\gamma = \arg \max_{\gamma \in N_\gamma(v)} (N_\gamma(v)) | class_\gamma \leq class_{bin_x}$ 
9       $bin_x \leftarrow \gamma$ 
10     set  $\gamma$  : reassigned
11     set  $bin_x$  : used
12 end while

```

---

Fig. 6. Algorithm of the Balance Module

## 7 Experimentation

There is no external method of comparison, because the problem addressed is a new rich BPP. Besides, the evaluation must be made in the context of a complete transportation system. Two versions were liberated to assist the necessities of a bottling company of products in two different instants of time and with a different constraint number: THSA [14] and ACS-ARH. The latter includes the algorithm proposed in this paper to solve the new rich BPP identified.

In THSA, when neither complete platform could be accommodated in the truck, these were segmented to improve the distribution of the load. However, the segmentation represents a bigger effort in the unloading. For this reason the segmentation was eliminated from ACS-ARH. In THSA some trucks could be programmed with a very small load, implying an excessive cost of the truck for so little demand. In ACS-ARH, the programming of vehicles is verified regarding to a minimum quantity of load; this constraint diminishes costs but reduces the percentage of satisfied demand, requiring a demand threshold from the user to determine the validity of a programming. The previous differences between THSA and ACS-ARH obstruct the right evaluation, so the second difference was eliminated.

The algorithm was developed in C# language and it was executed for two minutes, pointing out that the company needed 180 minutes approximately in order to solve the

problem manually. For the experimentation 214 real world instances were used. The instances were identified by the order programming date.

A sample of the algorithmic performance can be observed in table 4, which shows that ACS-ARHC obtains a reduction of two trucks on the average regarding to THSA. Another important discovery reveals that the percentage of satisfied demand is very similar in both algorithms. It is important to mention that in THSA the assignment unit is the bed and in ACS-ARHC the unit is the platform, which is bigger and therefore more difficulty to distribute. For the above-mentioned, it can consider that the deterministic algorithm DiPro, which is part of the system ACS-ARHC, works in a satisfactory way the accommodation of the products in the bins of the vehicles: it obtains percentages of demand satisfaction inside the limits and in a reasonable computed time (the company proposed ten minutes but two minutes were enough).

**Table 4.** Solution of instances of a Company Distributor of Bottled Products

Instances	THSA		ACS-ARHC	
	%Demands Satisfied	Used Vehicles	%Demands Satisfied	Used Vehicles
06/12/2005	99.48	5 *	97.66	4 *
09/12/2005	95.93	6 **	100	5
12/12/2005	95.53	3 **	99.06	4 *
01/01/2006	100	5	97.81	4 *
03/01/2006	100	4	100	3
07/02/2006	98.35	7 *	99.21	5 *
13/02/2006	98.43	6 *	99.52	4 *
06/03/2006	100	5	97.63	3 *
09/03/2006	99.30	6 *	98.65	4 *
22/04/2006	98.87	7 **	98.28	5 *
14/06/2006	98.33	7 **	97.83	6 *
04/07/2006	99.70	6 *	100	3 *
<b>Average (All Instances)</b>	98%	6	97.45%	4

\* One tour cancelled by the restriction of minimum load of the vehicle

\*\* Two tour cancelled by the restriction of minimum load of the vehicle

The number of cancelled tours has little impact on the percentage of unsatisfied demand (2.55%). In order to increase the satisfaction of the clients, we are developing a post processing procedure to accommodate the remaining products. In this method some constrains will be simplified.

## 8 Conclusions and Future Work

In this work the load accommodation into vehicles was approached as a part of a bigger rich problem: the product transportation. The deterministic algorithm DiPro is proposed for the construction of load solutions, it interacts with a metaheuristic algorithm developed to plan routes and schedules. The result of the experimentation shows that both algorithms diminish the number of vehicles to use and satisfy the

demand inside the limits established by the company. All instances were executed in a reasonable time for users that make transportation plans.

The quality of the solutions obtained with the deterministic algorithm DiPro is affected by the increment in the number of constraint and the conditions that are derived from the problem. For this reason, we propose as a future work the development of a hybrid metaheuristic algorithm that takes advantage of the strategies proposed in this research.

## References

1. Toth, P., Vigo, D.: The Vehicle Routing Problem. In: Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia (2001)
2. Pisinger, D., Ropke, S.: A General Heuristic for Vehicle Routing Problems, tech. report, Dept. of Computer Science, Univ. Copenhagen (2005)
3. OR/MS Today: Vehicle Routing Software Survey, United States. Institute for Operations Research and the Management Sciences (June 2006)
4. Cordeau, J.F., Laporte, G., Savelsbergh, M.W.P., Vigo, D.: Short-Haul Routing(submitted)
5. Gendreau, M., Iori, M., Laporte, G., Martello, S.: A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints, Networks (to appear)
6. Balgoh, J., Békési, J., Galambos, G., Reinelt, G.: Lower Bound for The On-line Packing Problem with Restricted Repacking. In: 29th Hungarian Conference of Mathematics, Physics and Computer Science (2005)
7. Herrera, J.: Development of a methodology based on heuristics for the integral solution of routing, scheduling and loading problems on the distribution and products delivery process. Master's thesis, Posgrado en Ciencias de la Computación, Instituto Tecnológico de Ciudad Madero, México (2006)
8. Kang, J., Park, S.: Algorithms for Variable Sized Bin Packing Problem. Proc. Operational Research 147, 365–372 (2003)
9. Epstein, L.: Online Bin Parking with Cardinality Constraints. In: Proc. 13th European Symposium on Algorithms (2005)
10. Chan, W., Chin, F.Y.L., Ye, D., Zhang, G., Zhang, Y.: Online Bin Parking of Fragile Objects with Application in Cellular Networks, tech. report, Hong Kong RGC Grant HKU5172/03E (2005)
11. Verweij, B.: Multiple Destination Bin Packing, tech. report, Algorithms and Complexity in Information Technology (1996)
12. Manyem, P.: Bin Packing and Covering with Longest Items at The Bottom: Online Version, ANZIAM J.43 (E), pp. E186–E232, Australia. Mathematical Soc. (2002)
13. Correia, I., Gouveia, L., Saldanha da, G.F.: Solving the Variable Size Bin Packing Problem with Discretized Formulations, Centro de Investigacao Operacional (CIO) - Working Paper (2006)
14. Cruz, L., González, J., Romero, D., Fraire, H., Rangel, N., Herrera, J., Arrañaga, B., Delgado, F.: A Distributed Metaheuristic for Solving a Real-World Scheduling-Routing-Loading Problem. In: Stojmenovic, I., Thulasiram, R.K., Yang, L.T., Jia, W., Guo, M., de Mello, R.F (eds.) ISPA 2007. LNCS, vol. 4742, pp. 68–77. Springer, Heidelberg (2007)

# On the Performance of Deterministic Sampling in Probabilistic Roadmap Planning

Abraham Sánchez L., Roberto Juárez G.<sup>†</sup>, and Maria A. Osorio L.

Facultad de Ciencias de la Computación, BUAP  
14 Sur esq. San Claudio, CP 72570  
Puebla, Pue., México

{[asanchez](mailto:asanchez@cs.buap.mx), [aosorio](mailto:aosorio@cs.buap.mx)}@cs.buap.mx, <sup>†</sup>[bein82@hotmail.com](mailto:bein82@hotmail.com)

**Abstract.** Probabilistic Roadmap approaches (PRMs) have been successfully applied in motion planning of robots with many degrees of freedom. In recent years, the community has proposed deterministic sampling as a way to improve the performance in these planners. However, our recent results show that the choice of the sampling source - pseudo-random or deterministic- has small impact on a PRM planner's performance. We used two single-query PRM planners for this comparative study. The advantage of the deterministic sampling on the pseudo-random sampling is only observable in low dimension problems. The results were surprising in the sense that deterministic sampling performed differently than claimed by the designers.

## 1 Introduction

A recent trend in motion planning has been the development of randomized planners [1], [2]. The main tradeoff for using randomization is that these planners are more efficient than their algebraic counterparts and can handle large degree of freedom (dof) problems, but at expense of completeness. Even though randomized planners are not complete<sup>1</sup>, a notion of probabilistic or asymptotic completeness has been established for many of them (see for example [3]).

While an algebraic planner would be overwhelmed by the prohibitive cost of computing an exact representation of the free configuration space ( $\mathcal{F}$ ), defined as the collision-free subset of configuration space ( $\mathcal{C}$ ), a PRM (probabilistic roadmap) planner builds only an extremely simplified representation of  $\mathcal{F}$  [4]. A roadmap is a graph whose nodes are configurations sampled from  $\mathcal{F}$  according to a suitable probability measure and whose edges are simple collision-free paths.

PRM planners have been successful in motion planning of robots with many dofs, but sampling narrow passages in  $\mathcal{C}$  remains a challenge for PRM planners. The first PRM planners used uniform random sampling of  $\mathcal{C}$  to select the nodes to add to the graph. In recent years other approaches have been suggested, either to create more samples in difficult regions or to remedy certain disadvantages

---

<sup>1</sup> A motion planner is *complete* if it always produces a path when one exists, and returns failure when one does not.

of the random behavior [5], [6], [7]. There are several sophisticated sampling strategies that can solve this difficulty [8]. Many of these strategies require complex geometric operations that are difficult to implement in high-dimensional configuration spaces.

Most of the studies made to compare the performance of the PRM planners has taken place with multiple-query planners. In this work, we concentrate on various deterministic and pseudo-random sampling methods for constructing probabilistic roadmaps. Until now, we don't have knowledge about a comparative study with single-query approaches. We used two planners, the expansive planner proposed by Hsu [11], [12] and the SBL planner proposed by Sánchez et al. [13]. Nevertheless, Geraerts and Overmars have made an interesting study with free-flying objects in a three-dimensional space [14], but only in the multiple-query approach. Such objects have six dofs. The authors used the most simple local method that consists of a straight-line motion in configuration space. Another interesting study was made in [15], in this study they proposed the use of single-query and multiple-query approaches and different local methods.

In Section II we discuss the importance of the sampling source. Section III presents an overview of the expansive and the SBL planners used for our comparative study. Several experimental comparisons are made in Section IV to compare the performance of both planners by using pseudo-random or deterministic sampling. Finally, conclusion and future work are presented in Section V.

## 2 The Importance of the Sampling Source

In general, to sample a configuration, a PRM planner needs both a probability measure  $\pi$  and a source  $S$  of random or deterministic numbers. The planner uses  $S$  to sample a point from a unit hypercube of suitable dimensionality and then maps the point into  $\mathcal{C}$  according to  $\pi$ . The most common source used in PRM planners is the pseudo-random source.

A pseudo-random number generator (PRNG) is an algorithm that uses arithmetic to generate a sequence of numbers that approximate the properties of random numbers. The sequence is not truly random because it is completely determined by a relatively small set of initial values, called the PRNG's state. Although sequences that are closer to truly random can be generated using hardware random number generators; pseudo-random numbers are important in practice for simulations (e.g., of physical systems with the Monte Carlo method), and are central in the practice of cryptography.

Most pseudo-random generator algorithms produce sequences which are uniformly distributed by any of several tests. Common classes of these algorithms are linear congruential generators, lagged Fibonacci generators, linear feedback shift registers and generalised feedback shift registers. Recent instances of pseudo-random algorithms include Blum Blum Shub, Fortuna, and the Mersenne twister.

Monte Carlo methods were mainly developed in the 1940s, by mathematicians and scientists. The term "Monte Carlo" was coined for these methods as a code word, by Von Neumann and Ulam suggesting the probabilistic nature of

these methods. Pseudo-random numbers are used in these simulations because they mimic the behavior of “real” random numbers. However, there are many Monte Carlo applications that do not really require randomness, but instead need numbers that uniformly cover the sample space. To meet these different requirements, quasi-random numbers have been developed. These are numbers that are very evenly distributed, but do not behave like truly random numbers. In fact, for certain problems one obtains deterministic, not probabilistic, bounds for Quasi-Monte Carlo methods.

Building on the quasi-Monte Carlo sampling literature, Lavalle et al. [7] have developed deterministic variants of the PRM by using low-discrepancy and low-dispersion samples, including lattices. Deterministic sampling sequences have the advantages of classical grid search approaches, i.e. a lattice structure (that allows to easily determine the neighborhood relations) and a good uniform coverage of the  $\mathcal{C}$ . Deterministic sampling sequences applied to PRM-like planners are demonstrated in [5], [7] to achieve the best asymptotic convergence rate and experimental results showed that they outperformed random sampling in nearly all motion planning problems. The work presented in [8] for nonholonomic motion planning proposes the use of different low-discrepancy sequences: Sobol, Faure, Niederreiter. An interesting study about sampling techniques in the PRM framework has proposed recently by Geraerts and Overmars [6]. The achievements of sampling-based motion planners are mainly due to their sampling-based nature, not due to the randomization (usually) used to generate samples.

In [8], the author presented an empirical study about the uniformity of the low-discrepancy sequences used to generate samples in a deterministic way. This study shows that the dimension is a factor that affects the performance of the deterministic generators. These experiments consolidate two ideas: 1) With a larger base, a low-discrepancy sequence can present certain pathologies?, and 2) does the minimal size of a low-discrepancy sample have a better equidistribution properties than a pseudo-random sequence that grows exponentially with the dimension?

### 3 Expansive and SBL Planners

The most popular paradigm for sampling-based motion planning is the Probabilistic Roadmap Method (PRM) [2]. PRM is a general planning scheme building probabilistic roadmaps by randomly selecting configurations from the free configuration space ( $\mathcal{C}$ ) and interconnecting certain pairs by simple feasible paths. The method has been applied to a wide variety of robot motion planning problems with remarkable success. PRM planners have been originally designed for solving multiple-query or single-query problems.

While multi-query planners use a sampling strategy to cover the whole free-space, a single-query planner applies a strategy to explore the smallest portion of free-space ( $\mathcal{F}$ ) needed to find a solution path. For example, see the planners presented in [12], [11] and [13].

Hsu *et al.* introduced a planner for “expansive”  $\mathcal{C}$  in [11]. The notion of expansiveness is related to how much of  $\mathcal{F}$  is visible from a single free configuration or

connected set of free configurations. The expansive-space planner grows a tree from the initial configuration. Each node  $q$  in the tree has an associated weight, which is defined to be the number of nodes inside  $N_r(q)$ , the ball of radius  $r$  centered at  $q$ . At each iteration, it picks a node to extend; the probability that a given node  $q$  will be selected is  $1/w(q)$ , in which  $w$  is the weight function. Then  $N$  points are sampled from  $N_r(q)$  for the selected node  $q$ , and the weight function value for each is calculated. Each new point  $q'$  is retained with probability  $1/w(q')$ , and the planner attempts to connect each retained point to the node  $q$ .

The planner in [13] searches  $\mathcal{F}$  by building a roadmap made of two trees of nodes,  $T_i$  and  $T_g$ . The root of  $T_i$  is the initial configuration  $q_i$ , and the root of  $T_g$  is the goal configuration  $q_g$  (bi-directional search). Every new node generated during the planning is installed in either one of the two trees as the child of an already existing node. The link between the two nodes is the straight-line segment joining them in  $\mathcal{CS}$ . This segment will be tested for collision only when it becomes necessary to perform this test to prove that a candidate path is collision-free (lazy collision checking).

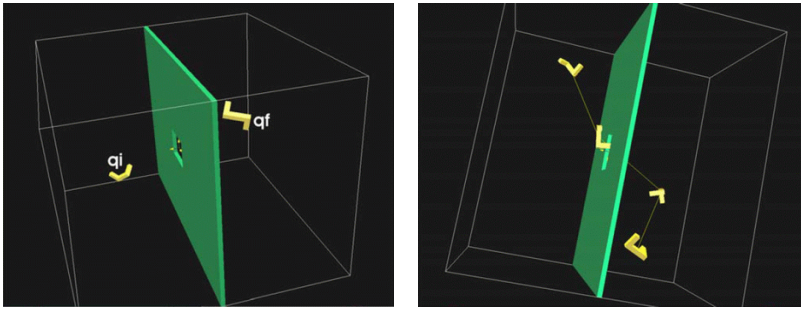
## 4 Experimental Results

The first papers on the PRM used uniform random sampling of the configuration space to select the nodes to add in a roadmap. In recent years, other uniform sampling approaches have been suggested to remedy certain disadvantages of the random behavior. The work proposed in [7] uses experimental problems with six dimensions. This kind of problems includes many applications of interest in robotics, but there are examples in robotics and computational biology in which dozens or hundreds of dimensions are needed. In addition, it is expected that the performance for high dimensional problems are negligible for most of them by using pseudo-random sampling or deterministic sampling.

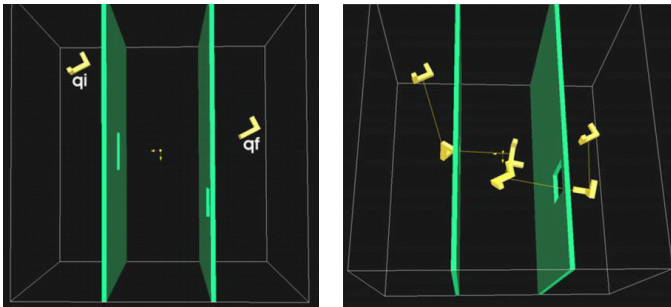
The main rationale for using deterministic sources is that they reduce the discrepancy or dispersion of the samples. However, the computational cost of achieving a fixed discrepancy or dispersion grows exponentially with the dimension of  $\mathcal{C}$  [4], [8]. The samples generated by a deterministic source are distributed evenly and regularly over  $[0, 1]^{dim(\mathcal{C})}$ . In PRM planning,  $N$  ( $N$  is the number of samples) is relatively small and the dimension of  $\mathcal{C}$  could be large (greater than six). This observation leads to large discrepancy and dispersion, even when a deterministic source is used.

Hence, the advantage that deterministic sources can possibly achieve over pseudo-random sources necessarily fades away as dimension of  $\mathcal{C}$  increases. To illustrate this, Figures 1, 2 and 3 compare the performance of the six sampling strategies (Random, Mersenne Twister, Sobol, Halton, Faure and Niederreiter) in configuration spaces of dimension six. In the figures, the configurations  $q_i$  and  $q_f$  respectively correspond to the start and goal positions of the problem. In all experiments we report the running time in seconds. Because the experiments are conducted under the same circumstances, the running time is a good indication of the efficiency of the technique.

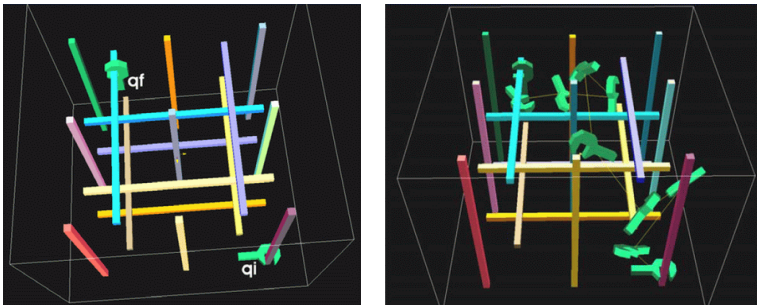




**Fig. 1.** The hole problem: A famous test scene, in which the moving object must rotate in a complicated way to get through the hole



**Fig. 2.** The hole II problem: A difficult version of the hole problem. The configuration space has three large open areas with three narrow winding passages between them.



**Fig. 3.** The wrench problem: This environment features a large moving object in a small workspace. The object is rather constrained at the start and the goal.

Tables [1](#), [2](#), [3](#), [4](#), [5](#), and [6](#) show the results of experiments performed on three different problems. In all experiment we used a simple local planner that consist of a straight-line motion in configuration space. We report statistics gathered

**Table 1.** Results of experiments performed for the hole problem with the expansive planner

Expansive planner						
Sampling method	Random	M. Twister	Sobol	Halton	Faure	Niederreiter
Nodes in the roadmap	641	658	909	804	538	740
Nodes in the path	14	14	16	16	14	16
Running time	38.02	41.03	56.67	50.70	33.87	48.06
Collisions checks	642764	688870	942342	838826	559195	762437

**Table 2.** Results of experiments performed for the hole problem with the SBL planner

SBL planner						
Sampling method	Random	M. Twister	Sobol	Halton	Faure	Niederreiter
Nodes in the roadmap	1083	1355	1868	1395	958	1312
Nodes in the path	19	21	23	22	21	22
Running time	1.16	1.52	2.16	1.83	1.48	5.03
Collision checks	13173	15578	17404	16730	14751	17029

**Table 3.** Results of experiments performed for the hole II problem with the expansive planner

Expansive planner						
Sampling method	Random	M. Twister	Sobol	Halton	Faure	Niederreiter
Nodes in the roadmap	1579	1666	2171	1567	1603	1117
Nodes in the path	19	20	22	20	21	20
Running time	97.22	105.36	134.75	101.12	104.63	70.94
Collisions checks	1472777	1572542	1981519	14386953	1482831	1013639

**Table 4.** Results of experiments performed for the hole II problem with the SBL planner

SBL planner						
Sampling method	Random	M. Twister	Sobol	Halton	Faure	Niederreiter
Nodes in the roadmap	19834	15920	22238	19038	19906	18981
Nodes in the path	61	56	64	62	51	52
Running time	46.79	40.26	73.12	66.24	74.75	111.16
Collision checks	123467	106007	141300	124700	109695	113979

from 50 independent runs when random sampling is used. The planners were implemented in Java 3D using the Voronoï Clip algorithm as collision checker.

The results show that many claims on efficiency of certain sampling approaches could not be verified. There was little difference between the various uniform sampling methods. One thing that is clear from this study is that a careful choice of techniques is important. Also, it is not necessarily true that a combination of good techniques and parameter choices results in optimal running times.

**Table 5.** Results of experiments performed for the wrench problem with the expansive planner

Expansive planner						
Sampling method	Random	M. Twister	Sobol	Halton	Faure	Niederreiter
Nodes in the roadmap	33	39	35	41	32	35
Nodes in the path	9	10	10	10	8	10
Running time	7.13	8.94	7.53	9.13	7.41	7.96
Collisions checks	22969	28362	23825	28793	23597	25043

**Table 6.** Results of experiments performed for the wrench problem with the SBL planner

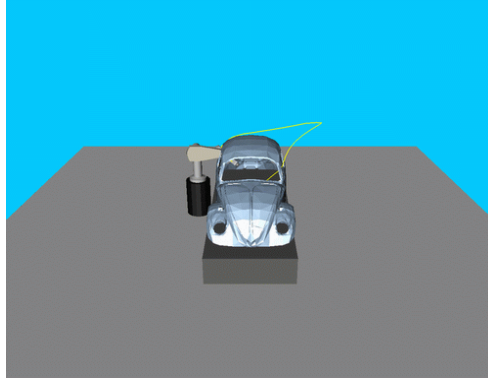
SBL planner						
Sampling method	Random	M. Twister	Sobol	Halton	Faure	Niederreiter
Nodes in the roadmap	3285	3720	3263	3260	2881	3825
Nodes in the path	32	33	31	36	29	34
Running time	17.68	21.58	20.12	21.50	16.36	34.31
Collision checks	50060	58764	53637	55284	40292	62245

The study also shows the difficulty of evaluating the techniques performance of the techniques. In particular the variance in the running time and the influence of certain bad runs were surprisingly large. Further research, in particular into adaptive sampling techniques, will be required to improve this. We compared techniques for free-flying objects and articulated robots. A major challenge is to create planners that automatically choose an appropriate combination of techniques based on scene properties or that learn the optimal settings while running.

An important difference between single-query and multiple-query planners was considered in [2]. Classic PRM was introduced as a pre-calculated data structure that could be used to quickly answer many queries in the same environment. Single-query versions of the PRM were introduced as a way to solve more difficult problems, see [12] and [13]. If we analyze the time needed to solve queries after constructing a roadmap, it can be said that there is an advantage regarding the time dedicated to pre-calculation procedures. However, the time invested is important only if a particular application uses many planning queries for the same environment.

An important issue that deserves more research is the idea of how to incrementally construct a data structure whose transformation velocity increases with the number of queries made. A single-query perspective could be used and part of its structure stored for future queries. The structure tends to adapt itself as more queries are provided. After several queries, the structure should have a small number of nodes; a higher number of queries will be answered more quickly due to this characteristic.

The following example considers a robot of nine degrees of freedom (see figure 4), one can observe in the table 7 that deterministic methods are sensible



**Fig. 4.** A mobile manipulator with nine degrees of freedom

**Table 7.** Results of experiments performed for the mobile manipulator problem with the SBL planner

Sampling method	SBL planner					
	Random	M. Twister	Sobol	Halton	Faure	Niederreiter
Nodes in the roadmap	211	741	748	Failed	801	627
Nodes in the path	19	23	22	Failed	22	21
Running time	1.12	1.45	1.51	Failed	1.94	1.56

to the dimension's problem. This detail verifies the empirical study made in [8] with respect to the uniformity of deterministic generators in high dimensions.

A crucial factor in the performance of PRM planners is how samples are generated. Sampling sequences should satisfy some requirements: the uniform coverage that can be incrementally improved as the number of samples increases, a locally controllable degree of resolution that allows to generate more samples at the critical regions.

## 5 Conclusion and Future Work

Sampling-based planners can successfully handle a large diversity of problems. The success of these planners in solving problems with many degrees of freedom and many obstacles can be explained by the fact that no explicit representation of the free configuration space ( $\mathcal{F}$ ) is required. The main operation of these planners is the ability of checking placements of the robot for collisions with obstacles in the environment, which can be efficiently performed by the current generation of collision checkers.

An ideal sampling strategy should create few samples that covers and connects  $\mathcal{F}$ . The smaller the number of samples, the less time is needed to connect those samples which is the most time-consuming step in the PRM. However, some

overlap between the regions that belong to the samples is required because this simplifies the process of creating connection between them. This can be achieved by creating a hybrid technique which filters out samples that do not contribute to extra coverage or maximal connectivity.

To speed up PRM, one promising direction is to design better sampling strategies (and perhaps connection strategies as well) by exploiting the partial knowledge acquired during roadmap construction and using this knowledge to adjust the sampling measure on-line to make it more effective.

The PRM tends to perform poorly when crucial configurations lie in and around narrow regions of the configuration space, which has been identified as the narrow passage problem. The probability of randomly guessing such a configuration can be very small, specially when the rest of the free space is large compared to these regions. Moreover, creating a set of configurations that covers a path that goes through the passage is not necessarily sufficient to solve the problem. The problem is only solved when all configurations in the set belong to the same connected component. Our experiments showed that this last criterion, which we call maximal connectivity, is much more difficult to satisfy than the coverage criterion, specially when we have to deal with a narrow passage.

The narrow passage problem can be tackled by incorporating a hybrid or adaptive sampling strategy that concentrates samples in difficult areas on one hand, and generates some samples in large open areas on the other hand. The use of the uniform sampling strategy is not a good choice for environments involving narrow passages. Another tactic is to employ a more powerful local planner.

The key of the success of contemporary algorithms to motion planning is the choice between randomness or heuristic strategies. These algorithms are based on sampling and consequently avoid the complexity of building configuration space obstacle representations, which other planners would have to implement. This aspect allows general purpose algorithms to evolve, while at the same time relegates the difficulties of analyzing the configuration space obstacles for collision detection algorithms.

## References

1. Barraquand, J., Latombe, J.C.: Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research* 10(6), 628–649 (1991)
2. Kavraki, L., Švetska, P., Latombe, J.C., Overmars, M.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12(4), 566–580 (1996)
3. Barraquand, K., Kavraki, L., Latombe, J.C., Motwani, R., Tsai-Yen, L., Raghavan, P.: A Random sampling scheme for path planning. *The International Journal of Robotics Research* 16(6), 759–774 (1997)
4. Hsu, D., Latombe, J.C., Kurniawati, H.: On the probabilistic foundations of probabilistic roadmap planning. *The International Journal of Robotics Research* 25(7), 627–643 (2006)
5. Branicky, M., LaValle, S., Olson, K., Yang, L.: Quasi-randomized path planning. In: *Proc. of the IEEE Robotics and Automation Conference*, pp. 1481–1487 (2001)

6. Geraerts, R., Overmars, M.: Sampling techniques for probabilistic roadmap planners. Technical Report UU-CS-2003-041, Utrecht University (2003)
7. Lavalle, S., Branicky, M., Lindemann, S.: On the relationship between classical grid search and probabilistic roadmaps. *International Journal of Robotics Research* 23(7-8), 673–692 (2004)
8. Sánchez, L.A.: Contribution à la planification de mouvement en robotique: Approches probabilistes et approches déterministes. PhD thesis, Université Montpellier II (2003)
9. Niederreiter, H.: Random number generation and Quasi-Monte Carlo methods. Society for Industrial and Applied Mathematics (1992)
10. Matousek, J.: Geometric discrepancy: An illustrated guide. Springer, Heidelberg (1999)
11. Hsu, D., Latombe, J.C., Motwani, R.: Path planning in expansive configuration spaces. *Int. J. of Computational Geometry and Applications* 9, 495–512 (1999)
12. Hsu, D.: Randomized single-query motion planning in expansive spaces. PhD thesis, Stanford University (2000)
13. Sánchez, G., Latombe, J.C.: On delaying collision checking in PRM planning: Application to multi-robot coordination. *The International Journal of Robotics Research* 21(1), 5–26 (2002)
14. Geraerts, R., Overmars, M.: A comparative study of probabilistic roadmap planners. In: Proc. of the Workshop on Algorithmic Foundations of Robotics, pp. 43–57 (2002)
15. Sánchez, L.A., Zapata, R., Lanzoni, C.: On the use of low-discrepancy sequences in non-holonomic motion planning. In: Proc. of the IEEE Robotics and Automation Conference, pp. 3764–3769 (2003)

# Hybrid Evolutionary Algorithm for Flowtime Minimisation in No-Wait Flowshop Scheduling

Geraldo Ribeiro Filho<sup>1</sup>, Marcelo Seido Nagano<sup>2</sup>,  
and Luiz Antonio Nogueira Lorena<sup>3</sup>

<sup>1</sup> Faculdade Bandeirantes de Educação Superior  
R. José Correia Gonçalves, 57  
08675-130, Suzano - SP - Brazil  
geraldorf@uol.com.br

<sup>2</sup> Escola de Engenharia de São Carlos - USP  
Av. Trabalhador São-Carlense, 400  
13566-590, São Carlos - SP - Brazil  
drnagano@usp.br

<sup>3</sup> Instituto Nacional de Pesquisas Espaciais - INPE/LAC  
Av. dos Astronautas, 1758  
12227-010, São José dos Campos - SP - Brazil  
lorena@lac.inpe.br

**Abstract.** This research presents a novel approach to solve m-machine no-wait flowshop scheduling problem. A continuous flowshop problem with total flowtime as criterion is considered applying a hybrid evolutionary algorithm. The performance of the proposed method is evaluated and the results are compared with the best known in the literature. Experimental tests show the superiority of the evolutionary hybrid regarding the solution quality.

## 1 Introduction

This paper considers the m-machine no-wait flowshop scheduling problem. In a no-wait flowshop, the operation of each job has to be processed without interruptions between consecutive machines, i.e., when necessary, the start of a job on a given machine must be delayed so that the completion of the operation coincides with the beginning of the operation on the following machine. Applications of no-wait flowshops can be found in many industries. For example, in steel factories, the heated metal continuously goes through a sequence of operations before it is allowed to cool in order to prevent defects in the composition of the steel. A second example is a plastic product that requires a series of processes to immediately follow one another in order to prevent degradation. Similar situations arise in other process industries such as the chemical and pharmaceutical. Hall and Sriskandarajah [1] give in their survey paper a detailed presentation of the applications and research on this problem and indicate that the problem with the objective of total or mean completion time is NP-Complete in the strong



sense even for the two-machine case. Considering that flowtime or completion time of a job is the same when the job is ready for processing at time zero and that minimizing total or mean completion time are equivalent criteria, some of the works on the no-wait problem with the objective of minimizing any of these criteria include Adiri and Pohoryles [2], Rajendran and Chaudhuri [3], Van der Veen and Van Dal [4], Chen et al. [5], Aldowaisan and Allahverdi [6], Aldowaisan [7], and Allahverdi and Aldowaisan [8].

Chen et al. [5] later develop a genetic algorithm and compare it with the heuristics of Rajendran and Chaudhuri [3]. Aldowaisan and Allahverdi [9] proposed six heuristics for the no-wait flowshop with the objective of minimizing the total completion time, which perform better than the heuristics of Rajendran and Chaudhuri [3] and Chen et al. [5]. Aldowaisan and Allahverdi [10] proposed simulated annealing (SA) and GA-based heuristics for the no-wait flowshop scheduling problem with the makespan criterion by incorporating a modified Nawaz-Enscore-Ham (NEH) heuristic (see Nawaz et al. [11]), based on a new insertion technique and pairwise procedure.

Fink and Vo $\beta$  [12] presented some construction methods and metaheuristics for the no-wait flowshop scheduling problem with the total flowtime criterion. Construction methods presented were the NN, the Chins, and the Pilot heuristics whereas metaheuristics investigated were steepest descent (SD), iterated steepest descent (ISD), SA, and TS algorithms. See Fink and Vo $\beta$  [12] for the details of construction methods and metaheuristics. The application of above heuristics for the no-wait flowshop scheduling problem were based on HotFrame, a heuristic optimization tool developed by Fink and Vo $\beta$  [13]. All heuristics have been applied to the benchmark suite of Taillard [14], originally generated for the unrestricted permutation flowshop sequencing problem. In addition, Fink and Vo $\beta$  [12] provided a detailed analysis of construction methods, different neighborhood structures embedded in SD, ISD, SA and TS algorithms. According to results, SA and reactive tabu search (RTS) algorithms generated better results with a 1000s CPU time in connection with shift (insert) neighborhood on the basis of initial solutions provided by Chins and Pilot-10 heuristics.

Recently Pan Q-K., et al. [15] presented a discrete particle swarm optimization (DPSO) to solve the no-wait flowshop scheduling problem with both makespan and total flowtime criteria. The main contribution of this study is due to the fact that particles are represented as discrete job permutations and a new position update method is developed based on the discrete domain. In addition, the DPSO algorithm is hybridized with the variable neighborhood descent (VND) algorithm to further improve the solution quality. Several speed-up methods are proposed for both the swap and insert neighborhood structures. The DPSO algorithm is applied to both 110 benchmark instances of Taillard [14] by treating them as the no-wait flowshop problem instances with the total flowtime criterion, and to 31 benchmark instances provided by Carlier [16], Heller [17], and Reeves [18] for the makespan criterion. For the makespan criterion, the solution quality is evaluated according to the reference makespans generated by Rajendran [19] whereas for the total flowtime criterion, it is evaluated with the optimal



solutions, lower bounds and best known solutions provided by Fink and Vo $\beta$  [12]. The computational results show that the DPSO algorithm generated either competitive or better results than those reported in the literature. Ultimately, 74 out of 80 best known solutions provided by Fink and Vo $\beta$  [12] were improved by the VND version of the DPSO algorithm. Based on the literature examination we have made, the aforementioned metaheuristic presented by Pan Q-K., et al. [15] yields the best solutions for total flowtime minimization in a permutation no-wait flowshop. In this paper, we address the generic m-machine no-wait flowshop problem with the objective of minimizing total flowtime. We propose a new Hybrid Evolutionary metaheuristic Algorithm and compare with the heuristic of Pan Q-K., et al. [15].

## 2 Evolutionay Heuristic

We have used in this work a Evolutionary Heuristics (EH) based on classic Genetic Algorithms. The chromosome representation used in EH was a  $n$  positions array, were each position indicates a task in the solution schedule. The population size was fixed in 200 individuals empirically, to make room for good individuals in the initial population altogether with randomly generated individuals to give some degree of diversification.

As the quality of the individual in the initial population has great importance in the evolutionary strategies, we have tried to create such quality individuals with a variation of the heuristic called NEH presented by Nawaz et al. [11]. The original form of NEH initially sorts a set of  $n$  tasks according to non-descending values of the sum of task processing times by all machines. The two first tasks in the sorted sequence are scheduled to minimize the partial flow time. The remaining tasks are then sequentially inserted into the schedule in the position that minimizes the partial flow time. In the variation used in this work, after the sort, the first two tasks to be scheduled were randomly taken. The very first individual inserted into the population was generated by NEH, the variation of NEH was used to generate other individuals in a number given by

$$\min \left( \frac{n * (n - 1)}{4}, \frac{200}{2} \right), \quad (1)$$

and the remain part of the initial population was filled with randomly generated schedules.

The evaluation of the individuals was made by the minimization of the total flow time for no-wait flowshop. The individual insertion routine kept the population sorted, and the best individual, the one with the lowest total flow time, occupied the first position in the population. The insertion routine was also responsible for maintain only one copy of each individual in the population.

Twenty five new individuals were created by iteration, and possibly inserted into the population. The stop condition used was the maximum of 200 iterations or 20 consecutive iterations with no new individuals being inserted. All these parameters had its values chosen empirically as result of tests.

A new individual generation was made by randomly selecting two parents, one from the best 20% of the population, called the base parent, and the other from the entire population, called the guide parent. A crossover process known as Block Order Crossover (BOX), presented by Syswerda [20], was applied to both parents, generating a single offspring by copying blocks of genes from the parents. In this work the offspring was generated with 70% of its genes coming from the base parent. Several other recombination operators are studied and empirically evaluated by Cotta and Troya [21]. Investigation regarding position-oriented recombination operators is also possible in further studies.

After the crossover, the offspring had a probability of 70% to be improved by a local search procedure called LS1, shown in Figure 1. This procedure used two neighborhood types: permutation and insertion. The permutation neighborhood around an individual was obtained by swapping every possible pair of chromosome genes, producing  $n * (n - 1)/2$  different individuals. The insertion neighborhood was obtained by removing every gene from its position, and inserting it in each other position in the chromosome, producing  $n * (n - 1)$  different individuals. Both, the improvement probability and the number of genes coming from the base parent to the offspring were parameters which values were taken after several tests.

```

Procedure LS1(current_solution)
Begin
  cs = current_solution;
  stop = false;
  While (stop == false) do Begin
    P = Permutation_neighborhood(cs);
    sp = First s in P that eval(s) < eval(cs), or eval(s) < eval(t) for all t in P;
    I = Insertion_neighborhood(cs);
    si = First s in I that eval(s) < eval(cs), or eval(s) < eval(t) for all t in I;
    If (eval(sp) < min(eval(si), eval(cs))) then
      cs = sp;
    else
      If (eval(si) < min(eval(sp), eval(cs))) then
        cs = si;
      else
        stop = true;
  End;
Return cs;
End

```

**Fig. 1.** Pseudo-code for the LS1 Local Search Procedure

The new individual was then inserted into the population in the position relative to its evaluation, shifting ahead the subsequent part of the population, and therefore removing the last, and worst, individual.

At the end of the EH iterations, the very first individual in the population was considered as the best solution found so far.

To evaluate the EH, tests were made using the first Taillard [14] instances considered as no-wait flowshop, with  $n=20$  tasks and  $m=5, 10$  and  $20$  machines.

The code was written in C language and ran in a Pentium IV, 3.0 GHz, 1Gb RAM computer. The EH ran 10 times for each instance and the EH has obtained exactly the same as best solution values found so far, compared with the work of Fink and Voß [12] and Pan et al. [15].

The preliminary tests with the Taillard next instances, with 50 and 100 tasks, were showing not so promising results. Therefore, we have adapted the EH to a hybrid form using the Cluster Search (CS) algorithm, following described.

### 3 Clustering Search

The metaheuristic Clustering Search (CS), proposed by Oliveira and Lorena [22,23], consists of a solution clustering process to detect supposedly promising regions in the search space. The objective of the detection of these regions as soon as possible is to adapt the search strategy. A region can be seen as a search subspace defined by a neighborhood relation.

The CS has an iterative clustering process, simultaneously executed with a heuristic, and tries to identify solution clusters that deserve special interest. The regions defined by these clusters must be explored, as soon as they are detected, by problem specific local search procedures. The expected result of more rational use of local search is convergence improvement associated with reduction of computational effort.

CS tries to locate promising regions by using clusters to represent these regions. A cluster is defined by a triple  $G = (C, r, \beta)$  where  $C$ ,  $r$  and  $\beta$  are, respectively, the center, the radius of a search region around the center, and a search strategy associated with the cluster.

The center  $C$  is a solution that represents the cluster, identify its location in the search space, and can be changed along the iterative process. Initially the centers can be obtained randomly, and progressively tend to move to more promising points in the search space. The radius  $r$  defines the maximum distance from the center to consider a solution being inside the cluster. For example, the radius  $r$  could be defined as the number moves to change a solution into another. The CS admits a solution to be inside of more than one cluster. The strategy  $\beta$  is a procedure to intensify the search, in which existing solutions interact with each other to create new ones.

The CS consists of four components, conceptually independent, with different attributions: a metaheuristic (ME), an iterative clustering process (IC), a cluster analyzer (CA), and a local optimization algorithm (LO).

The ME component works as a full time iterative solution generator. The algorithm is independently executed from the other CS components, and must be able to continuously generate solutions for the clustering process. Simultaneously, the clusters are maintained as containers for these solutions. This process works as a loop in which solutions are generated along the iterations.

The objective of the IC component is to associate similar solutions to form a cluster, keeping a representative one of them as the cluster center. The IC is implemented as an online process where the clusters are feed with the solutions

produced by the ME. A maximum number of clusters is previously defined to avoid unlimited cluster generation. A distance metric must be defined also previously to evaluate solutions similarity for the clustering process. Each solution received by IC is inserted into the cluster having the center most similar to it, causing a perturbation in this center. This perturbation is called assimilation and consists of the center update according to the inserted solution.

The CA component provides an analysis of each cluster, at regular time intervals, indicating probable promising clusters. The so called cluster density  $\lambda_i$  measures the  $i$ -th cluster activity. For simplicity,  $\lambda_i$  can be the cluster's number of assimilated solutions. When  $\lambda_i$  reach some threshold, meaning that ME has produced a predominant information model, the cluster must be more intensively investigated to accelerate its convergence to better search space regions. CA is also responsible for the removal of low density clusters, allowing new and better clusters to be created, while preserving the most active clusters. The clusters removal does not interfere with the set of solutions being produced by ME, as they are kept in a separate structure.

Finally, the LO component is a local search module that provides more intensive exploration of promising regions represented by active clusters. This process runs after CA has determined a highly active cluster. The local search corresponds to the  $\beta$  element that defines the cluster and is a problem specific local search procedure.

## 4 Evolutionary Clustering Search for the No-Wait Flow Shop Problem

This research has used a metaheuristic called Evolutionary Clustering Search (ECS) proposed by Oliveira and Lorena [22,23] that combines Evolutionary Heuristics (EH) and Clustering Search, and has originally applied it to the No-Wait Permutation Flow Shop problem. The ECS uses an EH to implement the ME component of the CS and generate solutions that allow the exploration of promising regions. A pseudo-code representation of the ECS for the No-Wait Flowshop problem is shown in Figure 2.

As ECS has presented good performance in previous applications, and considering the accelerated convergence provided by CS when compared with pure, non hybrid, algorithms, the aim of this work was to attempt to beat the best results recently produced and found in the literature, even with larger computer times, characteristic of evolutionary processes. Seeking originality, this was another reason to apply CS in this research.

The Evolutionary Clustering Search for No-Wait Flow Shop (ECS-NWFS) presented in this work has used the EH presented in Section 2 with some different parameters values. The population was fixed in 500 individuals to make room for more individuals generated by the NEH variation with larger number of tasks. Therefore, the number of such individuals was given by Equation 1 adapted to the new population size. At each iteration, 50 new individuals were generated and possibly inserted into the population. The stop condition used was a maximum

```

Procedure ECS-NWFS()
Begin
  Initialize population P;
  Initialize clusters set C;
  While (stop condition == false) do Begin
    While (i < new_individuals) do Begin
      parent1 = Selected from best 40% of P;
      parent2 = Selected from the whole P;
      offspring = Crossover(parent1, parent2);
      Local_Search_LS1(offspring) with 60% probability;
      If (Insert_into_P(offspring))
        Assimilate_or_create_cluster(offspring, C);
      i = i + 1;
    End;
    For each cluster c in C
      If (High_assimilation(c))
        Local_Search_LS2(c);
    End;
  End;
End;

```

**Fig. 2.** Pseudo-code for the ECS algorithm

of 500 iterations or 20 consecutive iterations with no insertions. New individuals generation was made the same way, using base and guide parents with BOX crossover. All other parameter were kept the same and these new values were obtained empirically with several tests.

A cluster set initialization process was created to take advantage of the good individuals in the EH initial population. This routine scanned the population, from the best individual to the worst, creating new clusters or assimilating the individuals into clusters already created. A new cluster was created when the distance from the individual to the center of any cluster was larger than  $r = 0.85 * n$ , and the individual was used to represent the center of the new cluster. Otherwise, the individual was assimilated by the cluster with the closest center. The distance measure from an individual to the cluster center was taken as the number of swaps necessary to transform the individual into the cluster center. Starting from the very first, each element of the individual was compared to its equivalent in the cluster center, at the same position. When non coincident elements were found the rest of the individual chromosome was scanned to find the same element found in the cluster center, and make a swap. At the end, the individual was identical to the cluster center, and the number of swaps was considered as a distance measure. The clusters initialization process ended when the whole population was scanned or when a maximum of 450 clusters were created. The cluster radius and the maximum number of clusters were chosen after several tests.

The assimilation of an individual by a cluster was based on the Path Relinking procedure presented by Glover [24]. Starting from the individual chromosome, successive swaps were made until the chromosome became identical to the cluster center. The pair of genes chosen to swap was the one that more reduced, or less increased, the chromosome total flow time. At each swap the new chromosome

configuration was evaluated. At the end of the transformation, the cluster center was moved to (replaced by) the individual, or the intermediary chromosome, that has the best evaluation better than the current center. If no such improvement was possible, the cluster center remains the same.

The successfully inserted individuals were then processed by the IC component of ECS-NWFS. This procedure tried to find the cluster having the closest center and of which radius  $r$  the individual was within. When such cluster could be found, the individual was assimilated, otherwise, a new cluster was created having the individual as its center. New clusters were created only if the ECS-NWFS had not reached the clusters limit. Tests have shown that the number of cluster tends to increase at very first iterations, and slowly decrease as iterations continue and the ECS removes the less active clusters.

After the generation of each new individual by the EH, its improvement by LS1, and the insertion into the population, the ECS-NWFS executed its cluster analysis procedure, with two tasks: remove the clusters that had no assimilations in the last 5 iterations, and take every cluster that had any assimilation in the current iteration and ran it through a second local optimization procedure, called LS2 and shown in Figure 3.

```

Procedure LS2(current_solution)
Begin
  cs = current_solution;
  stop = false;
  While (stop == false) do Begin
    I = Insertion_neighborhood(cs);
    si = First s in I that eval(s) < eval(cs), or eval(s) < eval(t) for all t in I;
    If (eval(si) < eval(cs)) then Begin
      cs = si;
      P = Permutation_neighborhood(cs);
      sp = First s in P that eval(s) < eval(cs), or eval(s) < eval(t) for all t in P;
      If (eval(sp) < eval(cs)) then
        cs = sp;
      End else Begin
        Pnh = Permutation_neighborhood(cs);
        sp = Scan Pnh until sp is better than cs, or sp is the best in Pnh;
        If (eval(sp) < eval(cs))
          cs = sp;
        else
          stop = true;
      End;
    End;
  Return cs;
End

```

**Fig. 3.** Pseudo-code for the LS2 Local Search Procedure

Along the ECS-NWFS processing the best cluster was kept saved. At the end of the execution, the center of the best cluster found so far was taken as the final solution produced by the algorithm.

## 5 Computational Experiments with ECS-NWFS

The ECS-NWFS ran 10 times for each instance and Table 1 shows that, except by one single instance, the algorithm has obtained better results than the best found by Fink and Vo $\beta$  [12] and Pan et al. [15] for the Taillard instances with  $n=50$  and 100 tasks. The table also shows the average of 10 runs of the ECS-NWFS, and the percentage gap between this average and the best solution. The gap was very low, less than 0.5% for all test instances.

**Table 1.** New Best Known solution for Taillard's benchmarks with  $n=50$  tasks and  $m=5$  machines (Ta031-Ta040),  $m=10$  (Ta041-Ta050),  $m=20$  (Ta051-Ta060), and  $n=100$  tasks with  $m=5$  machines (Ta061-Ta070),  $m=10$  (Ta071-Ta080) and  $m=20$  (Ta081-Ta090), considered as No-Wait Permutation Flow Shop, for Flowtime minimisation

Inst	<i>F&amp;V</i>	<i>DPSO</i>	<i>ECS</i>	Avg	%Gap	Inst	<i>F&amp;V</i>	<i>DPSO</i>	<i>ECS</i>	Avg	%Gap
Ta031	76016	75682	<b>75668</b>	75674.2	0.01	Ta061	308052	303750	<b>303567</b>	304736.3	0.39
Ta032	83403	82874	<b>82874</b>	83028.2	0.19	Ta062	302386	297672	<b>296321</b>	297472.3	0.39
Ta033	78282	78103	<b>78103</b>	78112.3	0.01	Ta063	295239	291782	<b>290638</b>	291406.2	0.26
Ta034	82737	82467	<b>82359</b>	82370.6	0.01	Ta064	278811	277093	<b>274722</b>	275266.1	0.20
Ta035	83901	83493	<b>83476</b>	83476.0	0.00	Ta065	292757	289554	<b>288344</b>	288766.4	0.15
Ta036	80924	80749	<b>80671</b>	80685.1	0.02	Ta066	290819	287055	<b>285752</b>	286502.3	0.26
Ta037	78791	78604	<b>78604</b>	78628.8	0.03	Ta067	300068	297731	<b>296537</b>	297383.7	0.29
Ta038	79007	78796	<b>78672</b>	78707.7	0.05	Ta068	291859	287754	<b>285961</b>	286890.1	0.32
Ta039	75842	75825	<b>75639</b>	75709.1	0.09	Ta069	307650	304131	<b>303657</b>	304119.4	0.15
Ta040	83829	83430	<b>83430</b>	83507.0	0.09	Ta070	301942	298119	<b>296964</b>	297683.6	0.24
Ta041	114398	114051	<b>113908</b>	113984.4	0.07	Ta071	412700	409715	<b>407679</b>	408079.3	0.10
Ta042	112725	112427	<b>112180</b>	112255.0	0.07	Ta072	394562	390417	<b>389001</b>	389884.5	0.23
Ta043	105433	105345	<b>105345</b>	105357.1	0.01	Ta073	405878	402274	<b>402036</b>	402483.7	0.11
Ta044	113540	113367	<b>113201</b>	113273.2	0.06	Ta074	422301	417733	<b>417091</b>	417545.9	0.11
Ta045	115441	115295	<b>115295</b>	115335.8	0.04	Ta075	400175	397049	<b>395519</b>	396701.0	0.30
Ta046	112645	112459	<b>112459</b>	112481.2	0.02	Ta076	391359	387398	<b>386418</b>	387696.8	0.33
Ta047	116560	116631	<b>116444</b>	116453.7	0.01	Ta077	394179	391057	<b>390076</b>	390798.6	0.19
Ta048	115056	115065	<b>114945</b>	114973.5	0.02	Ta078	402025	399214	<b>397072</b>	398109.4	0.26
Ta049	110482	110367	<b>110367</b>	110371.8	0.00	Ta079	416833	413701	<b>411396</b>	412792.1	0.34
Ta050	113462	113427	<b>113427</b>	113427.0	0.00	Ta080	410372	406206	<b>406001</b>	406311.1	0.08
Ta051	172845	172981	<b>172740</b>	172774.1	0.02	Ta081	562150	558199	<b>556564</b>	557322.9	0.14
Ta052	161092	160836	<b>160739</b>	160745.5	0.00	Ta082	563923	561305	<b>559171</b>	560357.8	0.21
Ta053	160213	160104	<b>160104</b>	160263.0	0.10	Ta083	562404	560530	<b>558440</b>	559562.1	0.20
Ta054	161557	161690	<b>161492</b>	161549.0	0.04	Ta084	562918	559690	<b>557386</b>	558275.3	0.16
Ta055	167640	167336	<b>167081</b>	167081.0	0.00	Ta085	556311	551388	<b>550704</b>	551675.1	0.18
Ta056	161784	161784	<b>161460</b>	161558.9	0.06	Ta086	562253	558356	<b>557051</b>	558196.7	0.21
Ta057	167233	<b>167064</b>	167098	167099.7	0.02	Ta087	574102	571680	<b>568667</b>	569486.6	0.14
Ta058	168100	167822	<b>167822</b>	168020.9	0.12	Ta088	578119	574269	<b>572945</b>	574158.1	0.21
Ta059	165292	165207	<b>165207</b>	165215.5	0.01	Ta089	564803	560710	<b>557946</b>	559497.0	0.28
Ta060	168386	168386	<b>168386</b>	168386.0	0.00	Ta090	572798	568927	<b>566054</b>	567688.8	0.29

The quality of the initial population individuals, allied to diversity, and the performance of the local search routines, can be considered key factors for the quality of the final solutions.

Processing times had a large variation from 48 seconds for an instance with 50 tasks, to 1 hour and 3 seconds for an instance with 100 tasks.

## 6 Conclusion

The main objective of this work was apply CS to the No-Wait Permutation Flow Shop Scheduling Problem of original and inedited form. Experimental results presented in the tables have shown that the EH had comparable, and the ECS-NWFS method had superior performance compared with the best results found in the literature for the considered test problems, using the in-process inventory reduction, or minimization of the total flow time, as the performance measure. The computational effort was acceptable for practical applications.

The classic optimization problem of task schedule in No-Wait Flow Shop has been the object of intense research for decades. For practical applications this problem may be considered already solved, although, because of its complexity it still remains as a target for the search for heuristic and metaheuristic methods.

The research related in this paper was motivated by the above considerations, and have tried to rescue the essential characteristics of metaheuristic methods, balance between solution quality and computational efficiency, simplicity and implementation easiness.

## References

1. Hall, N.G., Sriskandarajah, C.: A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research* 44, 510–525 (1996)
2. Adiri, I., Pohoryles, D.: Flowshop/no-idle or no-wait scheduling to minimize the sum of completion times. *Naval Research Logistics Quarterly* 29, 495–504 (1982)
3. Rajendran, C., Chaudhuri, D.: Heuristic algorithms for continuous flow-shop problem. *Naval Research Logistics* 37, 695–705 (1990)
4. Van der Veen, J.A.A., Van Dal, R.: Solvable cases of the no-wait flowshop scheduling problem. *Journal of the Operational Research Society* 42, 971–980 (1991)
5. Chen, C.L., Neppalli, R.V., Aljaber, N.: Genetic algorithms applied to the continuous flow shop problem. *Computers and Industrial Engineering* 30, 919–929 (1996)
6. Aldowaisan, T., Allahverdi, A.: Total flowtime in no-wait flowshops with separated setup times. *Computers and Operations Research* 25, 757–765 (1998)
7. Aldowaisan, T.: A new heuristic and dominance relations for no-wait flowshops with setups. *Computers and Operations Research* 28, 563–584 (2000)
8. Allahverdi, A., Aldowaisan, T.: No-wait and separate setup three-machine flowshop with total completion time criterion. *International Transactions in Operational Research* 7, 245–264 (2000)
9. Aldowaisan, T., Allahverdi, A.: New heuristics for m-machine no-wait flowshop to minimize total completion time. *Omega* 32(5), 345–352 (2004)
10. Aldowaisan, T., Allahverdi, A.: New heuristics for no-wait flowshops to minimize makespan. *Computers and Operations Research* 30(8), 1219–1231 (2003)
11. Nawaz, M., Enscore, E.E., Ham, I.: A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega* 11, 91–95 (1983)
12. Fink, A., Voß, S.: Solving the continuous flow-shop scheduling problem by metaheuristics. *European Journal of Operational Research* 151, 400–414 (2003)
13. Fink, A., Voß, S.: HotFrame: a heuristic optimization framework. In: Voß, S., Woodruff, D. (eds.) *Optimization software class libraries*, pp. 81–154. Kluwer, Boston (2002)



14. Taillard, E.: Benchmarks for basic scheduling problems. *European Journal of Operational Research* 64, 278–285 (1993)
15. Pan, Q.K., Tasgetiren, M.F., Liang, Y.C.: A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. *Computers and Operations Research* (2007), doi: 10.1016/j.cor.2006.12.030
16. Carlier, J.: Ordonnancements a contraintes disjonctives. *RAIRO Recherche operationelle* 12, 333–351 (1978)
17. Heller, J.: Some numerical experiments for an MxJ flow shop and its decision-theoretical aspects. *Operations Research* 8, 178–184 (1960)
18. Reeves, C.: A genetic algorithm for flowshop sequencing. *Computers and Operations Research* 22, 5–13 (1995)
19. Rajendran, C.: A no-wait flowshop scheduling heuristic to minimize makespan. *Journal of the Operational Research Society* 45, 472–479 (1994)
20. Syswerda, G.: Uniform crossover in genetic algorithms. In: *International Conference on Genetic Algorithms (ICGA)*, Virginia, USA, pp. 2–9 (1989)
21. Cotta, C., Troya, J.M.: Genetic Forma Recombination in Permutation Flowshop Problems. *Evolutionary Computation* 6, 25–44 (1998)
22. Oliveira, A.C.M., Lorena, L.A.N.: Detecting promising areas by evolutionary clustering search. In: Bazzan, A.L.C., Labidi, S. (eds.) *Advances in Artificial Intelligence. LNCS (LNAI)*, pp. 385–394. Springer, Heidelberg (2004)
23. Oliveira, A.C.M., Lorena, L.A.N.: Hybrid evolutionary algorithms and clustering search. In: Grosan, C., Abraham, A., Ishibuchi, H. (eds.) *Hybrid Evolutionary Systems - Studies in Computational Intelligence. Springer SCI Series*, vol. 75, pp. 81–102 (2007)
24. Glover, F.: Tabu search and adaptive memory programing: Advances, applications and challenges. In: *Interfaces in Computer Science and Operations Research*, pp. 1–75. Kluwer Academic Publishers, Dordrecht (1996)

# Enhancing Supply Chain Decisions Using Constraint Programming: A Case Study

Luiz C.A. Rodrigues and Leandro Magatão

Federal University of Technology – Paraná (UTFPR), Mechanical Engineering Department,  
Av. Sete de Setembro 3165, 80230-901 Curitiba, PR, Brazil  
{lcar, magatao}@utfpr.edu.br

**Abstract.** A new approach is proposed to tackle integrated decision making associated to supply chains. This procedure enables reliable decisions concerning the set of order demands along a supply chain. This is accomplished by means of supply chain scheduling simulations, based on the use of Constraint Programming. The definition of time windows for all tasks poses as an indication that no infeasibility was found during supply chain analysis. Scheduling of orders along the supply chain is treated as a constraint satisfaction problem. It suffices to identify any feasible schedule to state that simulated decisions are acceptable. The main contribution of this work is the integration of Constraint Programming concepts within a decision-support system to support supply chain decisions.

**Keywords:** Supply chain, decision-support systems, constraint programming.

## 1 Introduction

The integration of supply chain (SC) decisions faces the same challenges seen in the integration of production planning at an industry. It can be stated that SC management and production planning – based on MRP/MRP II – consider the existence of several levels of decisions. Fig. 1 and 2 present the sequence or modules of decisions proposed for MRP [1] and SC management [2, 3], respectively. Due to these levels of decisions or modules, decisions (e.g., sales, distribution, production planning, etc.) are often treated as multi-objective problems; because each department (or person) involved sets its (or his/her) own goals. The problem is that quite often these personal goals set a confrontation among companies, departments, and persons.

The goal of this paper is to propose a new tool that addresses the opportunities that were pointed out by [4]: Integrate databases throughout the SC; integrate control and planning support systems; redesign organizational incentives; institute SC performance measurement; and expand view of the SC. Conflicting objectives due to autonomous management and lack of confidence are quite often seen both at SC and enterprise management. At this paper, it is advocated that this pitfall and most of the pitfalls presented by [4] are due to the existence of several levels of decisions. Under a rigorous point of view, sales, purchasing, distribution, production planning and scheduling, and any other SC decisions should be taken using an integrated

procedure, once they are inter-dependent problems. For instance, it is not up to the sales department to advocate about planning and scheduling decisions, but infeasible demands and due dates are of no use for production and will impact negatively the SC. Therefore, demand decisions should concern to the feasibility of the information sent for production planning and scheduling, or to any other decisions along the SC.

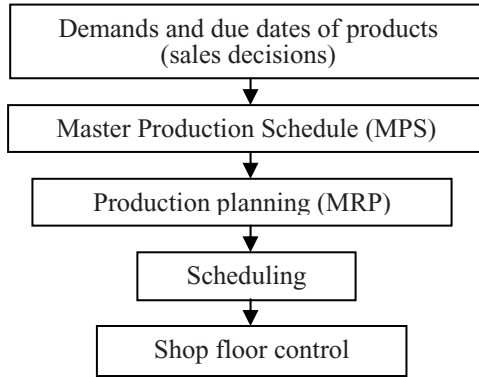


Fig. 1. Levels of decisions within MRP/MRP II [1]

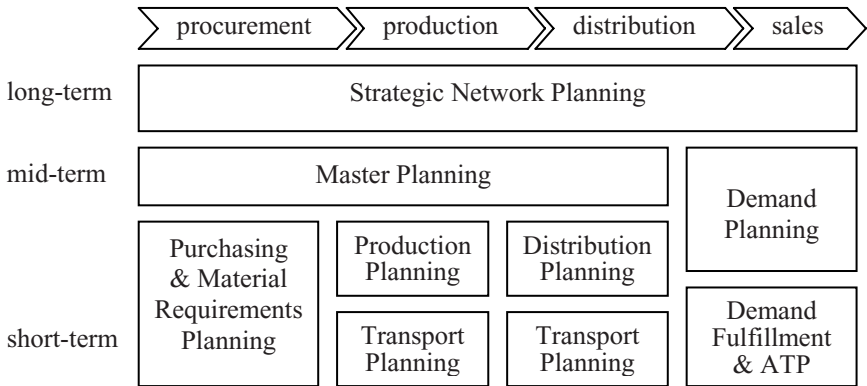


Fig. 2. Modules/levels of SC planning [2]

Literature review indicates that literature on SC improvement can be divided in five sorts of work: *i*) conceptual papers discuss collaboration [5, 6, 7], quick response [8, 9], and lean, agile and legile paradigms on SC [10, 11, 12]; *ii*) optimization papers discuss operations research approaches to solve specific SC problems, e.g. inventory [13, 14]; *iii*) simulation papers present multiagent and Petri net approaches to test SC strategies or decision support systems for SC [15, 16]; *iv*) information technology (IT) papers present the benefits – to SC – of e-commerce, EDI and other IT issues [17, 18]; and *v*) inventory policies [19, 20]. The use of Advanced Planning Systems (APS), discussed by [2], and operations research tools may identify optimal or near-

optimal SC solutions. But this “optimal” solution can make no sense on SC, because of the level of uncertainty associated to it. This level of uncertainty is due to short product lifecycle, high volatility, low predictability, and high level of impulse purchase [12]. Simulation tools can be used to improve SC performance, but it implies on significant data exchange. Whenever there is lack of confidence among partners in the SC, access to information and data exchange tend to be unavailable. Such scenario has stimulated the development of the proposed approach that has been previously applied as an enterprise decision support system [21]. It is not of the knowledge of the author of the present paper that Constraint Programming [22] has ever been used to support SC decisions.

In this work, it is proposed that any decisions should be made taking into account the feasibility of production. It is up to production planning department to set the optimal production plan and schedule. But any decision from other departments should consider production as a constraint satisfaction problem. In the proposed approach, for example, sales department will simulate SC scheduling, but only to check whether the demands and due dates for new orders are feasible. Once the proposed constraint satisfaction problem is much simpler than the scheduling problem itself, the goal of the proposed approach is to enable sales, purchasing, transport, and any other departments – including production planning – to ASAP answer to any customer consults or set feasible decisions upon request. In order to get a feasible and acceptable solution, the proposed simulation approach is based on two steps that are solved using constraint programming [22, 23]:

- *Preprocessing* procedure is intended to perform a capacity analysis allowing user intervention in order to achieve an acceptable plant loading.
- *Scheduling* procedure solves the problem. The purpose of preprocessing in this work is to prune the resulting scheduling problem.

Section 2 presents the proposed preprocessing procedure, while section 3 presents the scheduling procedure. In section 4, an example problem is proposed; and main results and capabilities of the proposed approach are presented. Section 5 concludes this paper.

## 2 Preprocessing Procedure

Changes due to disturbances in product demand, raw material availability, and equipment maintenance, can affect customer orders and consequently the production scheduling scenario. Therefore, preprocessing is intended to find out if product demands can be satisfied in terms of quantities and due dates, given the capacity of all SC plants and the time horizon of scheduling. The challenge for a sales department is to find out if the plant loading implied by a new demand of products is feasible, avoiding delayed customer orders and/or plant under utilization. Notice that equipment units and raw materials availability can affect the feasibility of planned orders as well. To complicate matters, plant capacity in multi-product/multipurpose facilities is only roughly defined as far as products can be manufactured following different paths using different equipment, due to the multipurpose nature of the plant.

The proposed approach structure is shown on Fig. 3. Each scenario is defined through a set of input data in order to get a compromise between plant loading and satisfaction of due dates. After the definition of input data, represented as white boxes in Fig. 3, preprocessing is performed in order to analyze the given scenario in terms of its potential feasibility. The main concept used in this approach is the concept of time window of a task. The time window of task  $i$  is a time interval defined by the earliest start time ( $EST$ ) and the latest finish time ( $LFT$ ) – or due date – for the production of task  $i$ . Any task must be performed within its time window in order to guarantee due date satisfaction of final products. Preprocessing has three main steps (Fig. 3):

- *Lots and latest finish times*: At this step, a backward procedure is performed in order to determine, through a backward consumed materials balance, the number of lots of each intermediate product and their respective due dates, referred as latest finish times ( $LFT$ ) in this work. This procedure is similar to the explosion procedure of MRP/MRP II systems but processing times are considered instead of lead times. Each product/intermediate has a latest production profile in order to satisfy the latest consumption profile of downstream tasks in the network. This profile is determined as the backward procedure is executed, and starts with the demand plan.
- *Earliest start times*: At this step, the user has to define a raw materials delivery plan. This plan is used to determine the earliest start times ( $EST$ ) of each lot of intermediate product through a forward consumed materials balance procedure. Once the earliest start time ( $EST$ ) and latest finish time ( $LFT$ ) of each intermediate product have been calculated, an initial time window for each lot is defined.  $EST$  (as well as  $LFT$ ) of any two successive lots of the same task must be shifted at least by the task processing time.
- *Capacity analysis and constraint propagation*: At this step, a capacity analysis is performed, using concepts of constraint propagation mechanisms [22, 23].

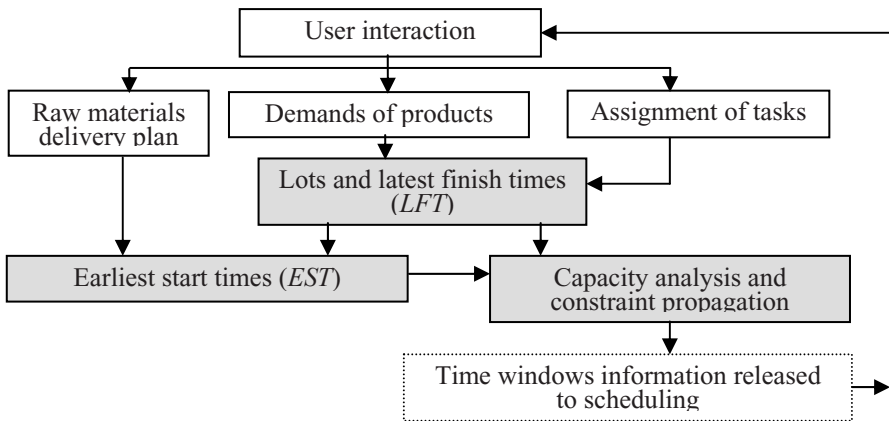


Fig. 3. Proposed approach structure

### 2.1 Capacity Analysis

There are two main ideas driving the ordering analysis: *i*) a time window of any task imposes a load on the assigned equipment unit; *ii*) the competition for the same unit within overlapping time intervals may launch reductions in time windows. Unit loading induced by time windows is analyzed using tools from constraint programming and artificial intelligence [22]. They allow finding out orderings among tasks that request the same equipment unit. Orderings can lead to reductions on time windows. The constraints that have to be satisfied during this analysis are: *(i)* each task must be processed inside given time windows; *(ii)* disjunctive constraints establish that each machine can process at most one task at a time. Table 1 indicates the used nomenclature for the analysis of orderings among tasks.

**Table 1.** Nomenclature

$EST_{i,k}$	Earliest start time of task $i$ on equipment unit $k$
$LFT_{i,k}$	Latest finish time of task $i$ on equipment unit $k$
$p_{i,k}$	Processing time of task $i$ on equipment unit $k$

Given a task  $C$  and a set of tasks  $\Omega$ , it is possible to identify whether this task precedes or is preceded by the set of tasks  $\Omega$ . To avoid the enumeration of all  $(2^n - 1)$  sets of  $n$  competing tasks on equipment unit  $k$ , a more efficient approach is adopted by using the concept of task interval [22]. A task interval is defined as the set of tasks  $\Omega$  defined by two tasks  $A$  and  $B$  that are processed in equipment unit  $k$ . Given a set of tasks processed in equipment unit  $k$ , then task  $i$  belongs to the set of tasks  $\Omega$  if its time window fits within the interval defined by the time windows of tasks  $A$  and  $B$ , as indicated by expressions (1) and (2). In this case, the number of tasks' intervals in unit  $k$  will be at most  $n^2$ . The  $EST$  for a set  $\Omega$  is taken as the minimum earliest start time of the tasks  $A$  and  $B$ , as indicated by expression (3). The  $LFT$  for a set  $\Omega$  is taken as the maximum latest finish time of the tasks  $A$  and  $B$ , as indicated by expression (4). The processing time of set  $\Omega$  is taken as the sum of tasks' processing times, as indicated by expression (5). If  $LFT_{\Omega,k} - EST_{C,k} < p_{\Omega,k} + p_{C,k}$ , then task  $C$  does not precede the entire set  $\Omega$ , so that expression (6) must be satisfied. Expression (7) must be satisfied if task  $C$  does not follow the entire set  $\Omega$ , that is, if  $LFT_{C,k} - EST_{\Omega,k} < p_{\Omega,k} + p_{C,k}$ . If  $LFT_{\Omega,k} - EST_{\Omega,k} < p_{\Omega,k} + p_{C,k}$ , then it means that  $C$  cannot be processed among tasks  $i$  belonging to set  $\Omega$ . In addition, if  $LFT_{\Omega,k} - EST_{C,k} < p_{\Omega,k} + p_{C,k}$  then set  $\Omega$  precedes task  $C$ . Therefore, a lower bound for  $EST_{C,k}$  is given by expression (8). Also in this case, due dates for any task  $i$  belonging to set  $\Omega$  must satisfy expression (9). The conclusion that set  $\Omega$  precedes  $C$  is also obtained if task  $C$  must follow any task  $i \in \Omega$  because  $EST_{C,k} + p_{C,k} > LFT_{i,k} - p_{i,k} (\forall i \in \Omega)$  and constraints in expressions (8) and (9) must hold.

If  $LFT_{\Omega,k} - EST_{\Omega,k} < p_{\Omega,k} + p_{C,k}$  and  $LFT_{C,k} - EST_{\Omega,k} < p_{\Omega,k} + p_{C,k}$  then task  $C$  precedes set  $\Omega$ . Therefore, an upper bound for  $LFT_{C,k}$  is given by expression (10).

Also in this case,  $EST_{C,k}$  for any task  $i$  belonging to set  $\Omega$  must satisfy expression (11). The conclusion that task  $C$  precedes set  $\Omega$  is also obtained if task  $C$  must precede any task  $i \in \Omega$  because  $LFT_{C,k} - p_{C,k} < EST_{i,k} + p_{i,k}$  ( $\forall i \in \Omega$ ) and constraints in expressions (10) and (11) hold.

$$EST_{i,k} \geq \min(EST_{A,k}, EST_{B,k}). \tag{1}$$

$$LFT_{i,k} \leq \max(LFT_{A,k}, LFT_{B,k}). \tag{2}$$

$$EST_{\Omega,k} = \min(EST_{A,k}, EST_{B,k}). \tag{3}$$

$$LFT_{\Omega,k} = \max(LFT_{A,k}, LFT_{B,k}). \tag{4}$$

$$p_{\Omega,k} = \sum_{i \in \Omega} p_{i,k}. \tag{5}$$

$$EST_{C,k} \geq \min_{i \in \Omega}(EST_{i,k} + p_{i,k}). \tag{6}$$

$$LFT_{C,k} \leq \max_{i \in \Omega}(LFT_{i,k} - p_{i,k}). \tag{7}$$

$$EST_{C,k} \geq EST_{\Omega,k} + p_{\Omega,k}. \tag{8}$$

$$LFT_{i,k} \leq LFT_{C,k} - p_{C,k} \quad \forall i \in \Omega. \tag{9}$$

$$LFT_{C,k} \leq LFT_{\Omega,k} - p_{\Omega,k}. \tag{10}$$

$$EST_{i,k} \geq EST_{C,k} + p_{C,k} \quad \forall i \in \Omega \tag{11}$$

### 2.2 Constraint Propagation

Orderings and storage restrictions can lead to reductions on time windows. Therefore, further analysis of orderings are triggered as a consequence of the propagation of these reductions on other time windows. Whenever reductions on time windows are identified, an analysis of recipe precedence constraints must be performed. Constraint propagation of materials flow balance follows the same mechanisms applied in the backward and forward explosion procedures, that is, any time window's reduction is propagated through the product recipe. The concepts of capacity analysis and constraint propagation presented within preprocessing procedure are also used as a scheduling tool in this work, but the proposal during preprocessing is to use these approaches to perform a preliminary feasibility analysis.

### 3 Scheduling Procedure

In general, the use of constraint programming to address scheduling problems leads to ordering decisions. Such decisions define an ordering relation between two lots (of

different tasks) competing for the same equipment unit. An ordering decision of this type can impose changes in both time windows: anticipation on *LFT* of the precedent lot and increment in *EST* of the succeeding lot. These changes in processing time windows can trigger further modifications on time windows for unscheduled lots through capacity analysis and propagation mechanisms. The search procedure outlined above can finish with a feasible solution or a dead end, if constraints of time windows cannot be satisfied. In the last situation it is necessary to backtrack and the search is reinitialized. In order to reduce the number of backtracks, orderings among pairs of lots are imposed in the resource with highest contention and the most constrained subset of lots competing for this resource.

Demand for an equipment unit, which are induced by processing time windows of lots assigned to it, can be evaluated using concepts such as cruciality function [24] or equipment unit slack [25]. The procedure proposed by [26] has been adopted in this work. This procedure presents an analysis based on individual demands of all lots imposed by their time windows. It suffices to identify individual demands at only four points to know all of their values. Individual demands will be equal to zero at points  $EST_{i,k}$  and  $LFT_{i,k}$ . At points  $EST_{i,k} + p_{i,k}$  and  $LFT_{i,k} - p_{i,k}$ , individual demands will be equal to  $\min\{1, p_{i,k} / (LFT_{i,k} - EST_{i,k} - p_{i,k})\}$ . Aggregated demands at an equipment unit will be equal to the sum of the individual demands of all lots processed at this unit along the scheduling horizon. Fig. 4 presents an example of how individual and aggregated demands are calculated, where the processing times of tasks *A* and *B* are 2 and 3 units of time, respectively.

Null or low competition means that it is almost assured that any task allocation will be possible. It seems that the number of backtracks is reduced when orderings are imposed in the resource with highest contention [23]. Therefore, equipment loading (that is, aggregated demands information) is used to guide branch-and-bound and, after an ordering is set, capacity analysis and constraint propagation tools are applied. This procedure is repeated throughout branch-and-bound search. When all orderings are defined at branch-and-bound, it can be stated that the solution is feasible.

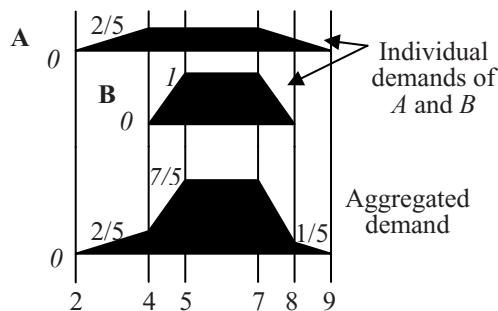


Fig. 4. Example of individual and aggregated demands



### 4 Solution of an Example

The proposed example is a “toy problem” representing the production process of three products. The purpose of using a toy problem is to make easier for the reader to understand the proposed approach and its benefits. The proposed approach has been successfully tested on real problems from process and manufacturing industries. Fig. 5 presents the recipes of products *ProA*, *ProB*, and *ProC*. Table 2 indicates assigned units to perform all tasks and their lot sizes. Demands and due dates of products are indicated on table 3. The purpose is to check if the proposed demands and due dates are feasible. Sales department wants to check if the addition of a demand of 150 units of product *ProB* – with due date at instant 56 – is feasible. It has been assumed that raw materials are available at the beginning of the scheduling horizon. But if the user wants, it is possible to simulate the impact of units or raw materials unavailability for a certain amount of time, as indicated on fig. 3.

Fig. 6 indicates the resulting time windows at the end of preprocessing procedure simulation. Right side of fig. 6 presents equipment loading information, while the left side of this figure indicates resulting time windows. Any interval of a time window that is presented in black indicates that the batch will necessarily occupy the assigned unit during that amount of time. On the other hand, any interval of a time window that is presented in white indicates that this interval is unavailable for the processing of the indicated batch. As a consequence of capacity analysis, it is possible to identify orderings and disjunctions among batches of different tasks. Given as pair of batches, there will be a disjunction if no ordering is identified between the two batches.

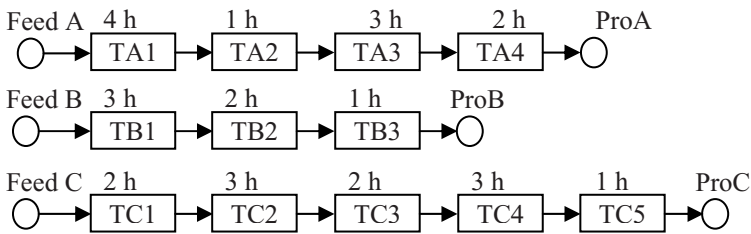


Fig. 5. Products recipes

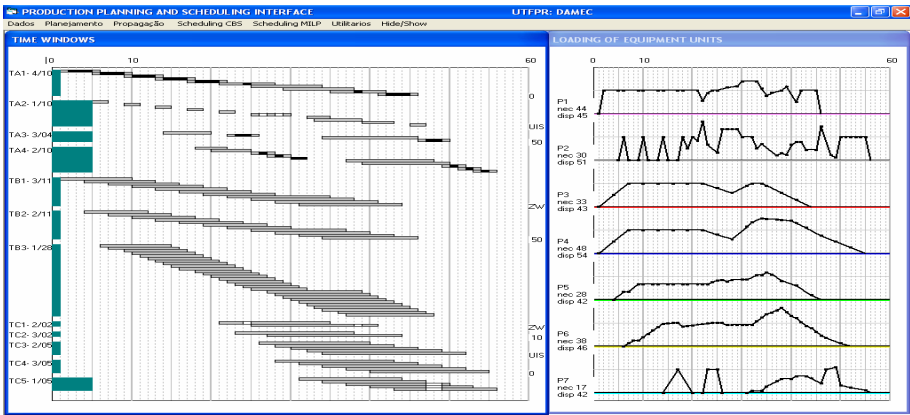
Scheduling is performed by imposing orderings to disjunctions located at the resource with the highest contention. This is done in an attempt to reduce the number of backtracks of the branch and bound procedure [23]. Whenever an ordering is imposed, capacity analysis and constraint propagation are performed once again. The goal of the scheduling simulation is only to check the feasibility of decisions on demands, task assignments, raw materials delivery date, etc. This goal is achieved when all disjunctions are ordered at branch and bound. Fig. 7 presents the time windows of a feasible schedule for the simulation of demands indicated on table 4. This schedule was obtained by setting orderings to 22 disjunctions. The proposed approach has always been able to solve industrial SC simulation problems in less than 2 minutes. The proposed example was solved in 8 s. on a Pentium 4 (3 GHz CPU and 2 MB of RAM memory).

**Table 2.** Assignments of tasks and batch sizes

Task	Storage policy	Assigned equipment unit	Batch size (ton)
TA1	NIS	P1	20
TA2	UIS	P1	40
TA3	FIS (50)	P2	20
TA4	UIS	P2	20
TB1	ZW	P3 and P4	40
TB2	FIS (50)	P4	15
TB3	UIS	P5	40
TC1	ZW	P5	40
TC2	FIS (100)	P6	15
TC3	UIS	P6	15
TC4	NIS	P7	50
TC5	UIS	P7	15

**Table 3.** Demands and due dates

Product	Demand (ton)	Due date (h.)
ProA	100	32
	90	56
ProB	270	32
	150	56
ProC	75	56



**Fig. 6.** Time windows at the end of preprocessing for demand presented on table 3

### 4.1 Enhancing SC Decisions

The ability to foresee the plant operation is very important for any industry due to the uncertainties found on world economy. Therefore, it will be up to SC and plant management to simulate any circumstances that are important whenever making a decision. Despite the fact that the presented case has been solved at plant level, *EST* and *LFT* information from time windows may be propagated forward and backward on the SC. In such situation, if any infeasibility is detected, an interactive procedure simulating new due dates, demands, or delivery dates could enhance SC decisions. SC may suffer from conflicting objectives because it may involve companies of different sizes, organizational structures, and different cultures. This situation can result in lack of confidence, self-protective behaviors, and inefficient information systems. The

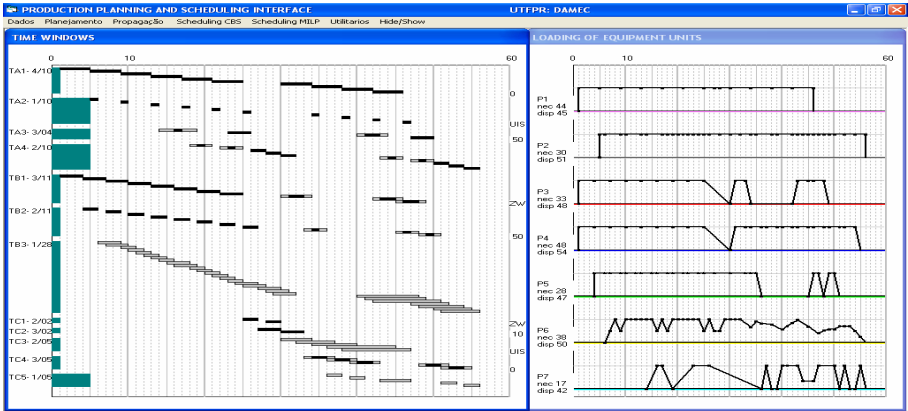


Fig. 7. Time windows of a feasible scheduling for demand presented on table 3

proposed approach is able to tackle all this SC problems. Unlike multiagent simulation systems that require extensive exchange of information along the chain, the proposed approach would work along the SC as if answering to customer inquiries on demands and due dates. Therefore, the customer along the SC will never access any confidential information from third parties. One of the benefits of this customer-salesperson relation set along the SC is that customer inquiries may be propagated along the SC. At the studied case, raw materials were assumed to be available at the beginning of the scheduling horizon. But time windows indicate that it will not be possible to start to process the first lot of task *TC1* before instant 21. Therefore, *FeedC* will only be needed short time before instant 21. If *FeedC* is available at the beginning of the scheduling horizon, it will be an unnecessary inventory impacting negatively on the costs of the SC. The rationale is that customer inquiries will be propagated backwards setting new customer inquiries along the SC. If this is implemented on a SC, significant lead times reductions may be achieved.

## 5 Conclusions

The goal of this paper is to propose a new tool that addresses the opportunities that had been pointed out by [4]. This decision-support tool is based on Constraint Programming and allows an unharmed propagation of information along the SC. In such a way, this tool implies in a new concept for database integration. The proposed tool also sets a unification of enterprise, production, and SC decisions. The industrial examples tested so far have indicated that the proposed approach has the potential to support real world production and SC decisions problems. The primary question now seems to be how to help the user to extract secure information from equipment loading profile. In order to generate reduced models it is important to select which equipment units, tasks, and periods are most relevant and have to be the focus of the

user's attention. ERP systems may reach an unprecedented level of decisions integration with the use of the proposed concepts. Sales and other SC decisions may be taken with the support of the proposed and other constraint programming tools.

## References

1. Orlicky, J.: *Material Requirements Planning*. McGraw-Hill, New York (1975)
2. Stadtler, H.: Supply chain management and advanced planning – basics, overview and challenges. *European Journal of Operational Research* 163, 575–588 (2005)
3. Meyer, H., Wagner, M., Rohde, J.: Structure of advanced planning systems. In: Stadtler, H., Kilger, C. (eds.) *Supply Chain Management and Advanced Planning – Concepts, Models Software and Case Studies*, pp. 99–104. Springer, Berlin (2002)
4. Lee, H., Billington, C.: Managing supply chain inventory: Pitfalls and opportunities. *Sloan Management Review* 33, 65–73 (1992)
5. Simatupang, T.M., Wright, A.C., Sridharan, R.: The knowledge of coordination for supply chain integration. *Business Process Management Journal* 8, 289–308 (2002)
6. Simatupang, T.M., Wright, A.C., Sridharan, R.: Applying the Theory of Constraints to supply chain collaboration. *Supply Chain Management: An Int. Journal* 9, 57–70 (2004)
7. Barrat, M.: Understanding the meaning of collaboration in the supply chain. *Supply Chain Management: An International Journal* 9, 30–42 (2004)
8. Perry, M., Sohal, A.S.: Quick response practices and technologies in developing supply chains: A case study. *Int. J. of Physical Distribution & Logistics Manag.* 30, 627–639 (2000)
9. Zairi, M.: Best practice in supply chain management: The experience of the retail sector. *European Journal of Innovation Management* 1, 59–66 (1998)
10. Christopher, M., Towill, D.R.: Supply chain migration from lean and functional to agile and customized. *Supply Chain Management: An International Journal* 5, 206–213 (2000)
11. Lin, C.T., Chiu, H., Chu, P.Y.: Agility index in the supply chain. *Int. J. Prod. Econ.* 100, 285–299 (2006)
12. Mason-Jones, R., Naylor, B., Towill, D.R.: Engineering for the leagile supply chain. *International Journal of Agile Management Systems* 2, 54–61 (2000)
13. Agrawal, V., Chao, X., Seshadri, S.: Dynamic balancing of inventory in supply chains. *European Journal of Operational Research* 159, 296–317 (2004)
14. Bertazzi, L., Paletta, G., Speranza, M.G.: Minimizing the total cost in an integrated vendor-managed inventory system. *Journal of Heuristics* 11, 393–419 (2005)
15. Raghavan, N.R.S., Roy, D.: A stochastic Petri net approach for inventory rationing in multi-echelon supply chains. *Journal of Heuristics* 11, 421–446 (2005)
16. Fleisch, E., Tellkamp, C.: Inventory inaccuracy and supply chain performance: A simulation study of a retail supply chain. *Int. J. Prod. Econ.* 95, 373–385 (2005)
17. Johnson, M.E., Whang, S.: E-business and supply chain management: An overview and framework. *Production and Operations Management* 11, 413–423 (2002)
18. Gunasekaran, A., Marri, H.B., McGaughey, R.E., Nebhwani, M.D.: E-commerce and its impact on operations management. *Int. J. Prod. Econ.* 75, 185–197 (2002)
19. Lee, H., Padmanabhan, V., Whang, S.: Bullwhip effect in supply chains. *Sloan Management Review* 38, 93–102 (1997)
20. Disney, S.M., Towill, D.R.: The effect of vendor managed inventory (VMI) dynamics on the Bullwhip effect in supply chains. *Int. J. Prod. Econ.* 85, 199–215 (2003)

21. Rodrigues, L.C.A., Riechi, J.L.S.: Capacity analysis tool to support sales decisions. In: 18th International Conference on Production Research, Salerno, Italy (2005)
22. Baptiste, P., Le Pape, C., Nuijten, W.: Constraint-based scheduling. Kluwer Academic Publishers, Norwell, Massachusetts (2001)
23. Rodrigues, M.T.M., Latre, L.G., Rodrigues, L.C.A.: Short-term planning and scheduling in multipurpose batch chemical plants: a multi-level approach. *Computers and Chemical Engineering* 24, 2247–2258 (2000)
24. Keng, N.P., Yun, D.Y.Y., Rossi, M.: Interaction sensitive planning system for job shop scheduling. In: *Expert syst. and intelligent manufacturing*, pp. 57–69. Elsevier, Amsterdam (1988)
25. ILOG: Scheduler 4.0 User's Manual. ILOG, Mountain View, USA (1997)
26. Beck, J.C., Davenport, A.J., Sitarski, E.M., Fox, M.S.: Texture-based heuristics for scheduling revisited. In: *Proc. of the American Association for Artificial Intelligence* (1997)

# Analysis of DNA-Dimer Distribution in Retroviral Genomes Using a Bayesian Networks Induction Technique Based on Genetic Algorithms

Ramiro Garza-Domínguez<sup>1</sup> and Antonio Quiroz-Gutiérrez<sup>2</sup>

<sup>1</sup> Centro de Tecnologías de Información, Universidad Autónoma del Carmen,  
Ciudad del Carmen, Campeche, Mexico  
rgarza@pampano.unacar.mx

<sup>2</sup> Facultad de Química, Universidad Autónoma del Carmen,  
Ciudad del Carmen, Campeche, Mexico  
aquiroz@pampano.unacar.mx

**Abstract.** Since DNA-dimer analysis has demonstrated to provide a very conserved pattern that has been suggested as a genome signature, in this paper we present a computational study of DNA-dimer distribution in a collection of Retroviral genomes. This analysis is based on two main steps: the generation of the target dataset, in this step, the DNA-dimer distribution variables are calculated and transformed to categorical data using Fuzzy Sets. And the induction of a Bayesian Network from the dataset. This induction technique is based on Genetic Algorithms. We have found interesting causal relationships between the DNA-dimer distribution variables and a set of chemical variables. These results could provide new directions in future Retroviral genomic investigations. The computational methodology presented in this paper has demonstrated to be an interesting tool for the study and the analysis of genomic sequences.

**Keywords:** Bioinformatics, Retroviral Genomes, DNA-dimer, Fuzzy Sets, Bayesian Networks.

## 1 Introduction

After the synthesis of DNA mono crystals late in the eighties, it was thought this polymer as a static and rigid one only randomly disturbed by mutations. However, the discovery of processes as transposition, hyper-mutation, genetic drive, gene conversion and the numerous arrangements of the genes involved in the immune response, shows that this molecule is far from being rigid or static [1]. From the biophysical point of view, studies of molecular structure and dynamics have also shown that this polymer is far from being a rigid or static molecule [1]. During the last three decades, different restrictions in the nucleotide sequence unable to be predicted by the Watson and Crick's model were demonstrated by different methods [1, 2]. Early in the sixties when biochemical methods of DNA-dimer frequency analysis were applied to samples of genomic DNA in many organisms, it was observed that

representation of some DNA-dimers deviated significantly from the statistically expected value. Susumu Ohno systematized the dimer under and over representation in which he called the rule of TA/CG deficiency and TG/CT excess, these restrictions appear to be determinant for some important facts in biology [1].

The study of DNA-dimer properties is of interest due to the fact that exists a relation between DNA-dimer statistical distribution and the basic conditions for DNA physicochemical stability [3], and also because it is possible that DNA-dimer distribution is related with a genetic signature useful for phylogenetic and taxonomical classification of species [1]. The Retroviridae is a family of RNA viruses that infect vertebrates. Retroviruses can cause a variety of diseases like sarcoma, leukemia and immunodeficiency [4]. Even when Retroviral genomes in general, are relatively small genomes, they constitute examples of extremely complex systems. They are able to transcript over twenty five different proteins, since they read the genetic message in quite different ways. HIV in special, has a mutation rate six orders of magnitude higher than most living organisms. Because the Retroviral complex dynamics, the study and identification of genomic patterns have become a challenge.

In this paper, a computational analysis of DNA-dimer distribution in a collection of Retroviral genomes is presented. This analysis is based on two main steps: the generation of the target dataset, in this step, the DNA-dimer distribution variables are calculated from the genomes, and transformed to categorical data using Fuzzy Sets. And the induction of a Bayesian Network from the dataset. This induction technique is based on Genetic Algorithms. With this strategy we pretend to find causal relationships between the DNA-dimer distribution variables and a set of chemical variables, in order to characterize the Retroviral genomes from the DNA-dimer perspective. In section 2, we present an overview of the Retroviridae family and the collection of viruses under study. In section 3, we describe the steps in the generation of the target dataset. In section 4, we describe the Bayesian Networks induction technique. In section 5, we present some of the experimental results that we have obtained. Finally, in section 6 we present the conclusions of this work.

## 2 The Retroviridae Family

The Retroviridae is a family of RNA viruses that infect vertebrates. Most RNA viruses enter the host cell and act as mRNA or are transcribed into mRNA. The Retroviridae are different, they carry with them a unique enzyme called reverse transcriptase. This enzyme is an RNA-dependent DNA polymerase that converts the viral RNA into DNA [4]. This viral DNA has unique sticky ends that allow it to integrate into the host's own DNA. Retroviruses can cause cancer in the cells they infect, genes called oncogenes can cause the malignant transformation of normal cells into cancer cells. Some retroviruses carry oncogenes in their genome. Another important feature is that some Retroviridae are cytotoxic to certain cells, blowing them up. The most notable is the human immunodeficiency virus which destroys the CD4 T helper lymphocytes it infects. This ultimately results in devastating immunodeficiency. Some retroviruses cause cancer directly by integrating an intact oncogene into the host DNA, these are called acute transforming viruses. Others cause cancer indirectly by activating a host proto-oncogene, these are called non-

acute transforming viruses [4]. The acute transforming viruses were discovered in 1911 when Peyton Rous injected cell free filtered material from a chicken tumor into another chicken. The chicken subsequently developed tumor. The causative agent is now known to be a Retrovirus called the Rous sarcoma virus [4]. The non-acute transforming viruses, activate host cell proto-oncogenes by integrating viral DNA into a key regulatory area.

	Virus	Genus
	Avian leukosis virus - RSA	Alpha
	Rous sarcoma virus	Alpha
	Ovine pulmonary adenocarcinoma virus	Beta
	Enzootic nasal tumour virus of goats	Beta
	Simian T-lymphotropic virus 1	Delta
	Human T-lymphotropic virus 2	Delta
	Walleye dermal sarcoma virus	Epsilon
	Spleen focus-forming virus	Gamma
	Feline leukemia virus	Gamma
	Human immunodeficiency virus 1	Lenti
	Equine infectious anemia virus	Lenti
	Avian endogenous retrovirus EAV-HP	Non
	Simian foamy virus	Spuma
	Human spumaretrovirus	Spuma

**Fig. 1.** A subset of the Retroviruses

By the mid 1970's, Retroviruses had been discovered in many vertebrate species, including Apes. The hypothesis that humans may also be infected with Retroviruses led to a search that ultimately resulted in the isolation of a Retrovirus from the cell lines and blood of patients with adult T-cell leukemia. This virus is called human T-cell leukemia virus. This virus has now been linked to a paralytic disease that occurs in the tropics called tropical spastic paraparesis [4]. In early 1980 a new epidemic was first noted that we now call the Acquired Immunodeficiency Syndrome. Investigators stimulated T-cell culture growth and were able to find RNA and DNA, suggesting a Retroviral etiology. The virus which was subsequently identified, was called the human immunodeficiency virus HIV, the cause of the world's most feared current epidemic [4].

Retroviruses are currently classified into seven genera [5]: Alpharetrovirus, Betaretrovirus, Deltaretrovirus, Epsilonretrovirus, Gammaretrovirus, Lentivirus and Spumavirus. As mentioned before we are interested in finding causal relationships between the DNA-dimer distribution variables and a set of chemical variables, in order to characterize the Retroviral genomes from the DNA-dimer perspective. For this experiment, we have selected all the Retroviral genomes available from the GenBank through the Entrez Documental Retrieval System. By the time of this writing, there are 55 Retroviral genomes. In figure 1 we show a subset of the 55 Retroviruses under study.



### 3 Calculating the Target Dataset

In order to calculate the target dataset, three main steps are necessary. The first step consists in calculate the DNA-dimer distribution variables from the genomes. In the second step the values of these variables are transformed to categorical data. Finally, some chemical variables are added to complete the dataset.

#### 3.1 DNA-Dimer Distribution

The DNA-dimer distribution pattern of a genome, is conformed by a two dimensional relation applied to the sixteen possible dimers. The first variable in this relation corresponds to the average inter space between all the occurrences of a dimer in the genome. The second variable corresponds to the frequency of the dimer in the genome. Because the genomes are of different length, it is necessary to convert the values of the distribution variables to percents. The target dataset consists in a table which we call, the intensity table. In this table, each row corresponds to a specific dimer of a certain Retroviral genome, and the columns correspond to properties of the dimer. The first two variables in the intensity table are the DNA-dimer distribution variables mentioned above. In order to apply the Bayesian Networks induction technique, it is necessary to convert the numeric values of the distribution variables to categorical data. In this case we have selected the Fuzzy Sets approach because its interesting features.

#### 3.2 Data Transformation Using Fuzzy Sets

Fuzzy logic is a form of mathematical logic based on partial truth values, it means that a truth value of a proposition can be a real number between 0 and 1. Fuzzy logic arose as an intent to model uncertainty or imprecision in knowledge [6]. The most basic components in Fuzzy logic are the Fuzzy sets. A Fuzzy set is a set with no clearly defined bounds, it means that it contains elements in partial grades of membership, contrary to the classic set theory based on crisp logic where each element belongs entirely or not to a set. Fuzzy sets discriminate in a better way and provide more information. The grade of membership of an element to a set represents the compatibility of that element with the idea that the set represents. Fuzzy sets are represented with a graph where the horizontal axis corresponds to the elements in the set, the vertical axis corresponds to the grade of membership to that set, and the curve defines how each element in the input space is mapped to a grade of membership. This curve is called the membership function. There are different types of membership functions, the four standard types are: Z, Lambda, Pi and S [6]. These functions have the advantage that they are easy to interpret and are computational efficient. A linguistic variable is a variable which combines multiple subjective categories that describe the same context, the possible values of a linguistic variable are called linguistic terms and are defined by Fuzzy sets. The main function of a linguistic variable is to translate a numeric value to a linguistic value by means of the Fuzzy sets.

In this case we have defined two linguistic variables, which we call Space and Frequency and represent the DNA-dimer distribution variables. These linguistic

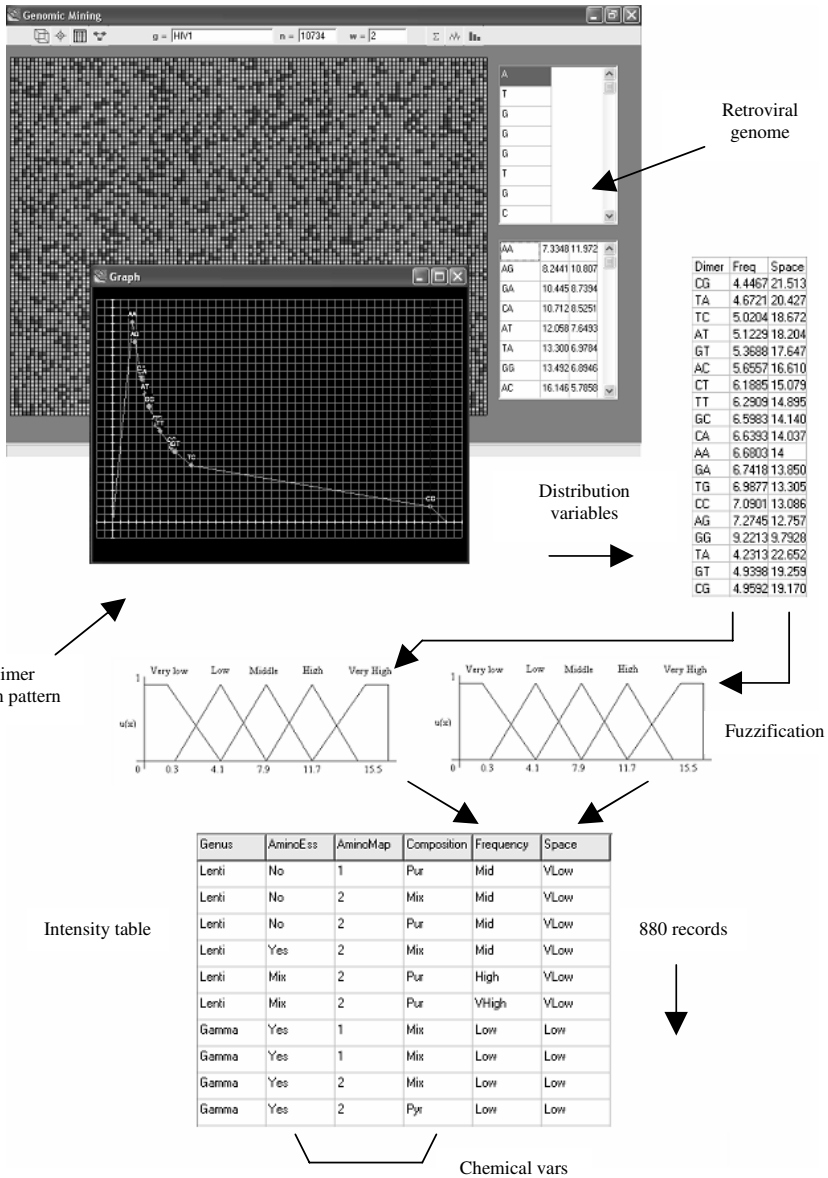


Fig. 2. Calculating the target dataset

variables are the mechanism to convert the numeric values from the intensity table to categorical values. To define the linguistic terms, we use five Fuzzy sets distributed between the minimum and maximum values relative to each variable. We use the Z and S membership functions for the terms in the extremes, and the Lambda membership function for the middle sets. With these linguistic variables it is possible

to convert the numeric values of the intensity table to categorical values, calculating the grade of membership for each value to the five Fuzzy sets. Then the linguistic term corresponding to the Fuzzy set with the highest grade of membership will be the new value.

### 3.3 Chemical Variables

The DNA molecule is composed by a sequence of bases, there are four types of bases represented by the letters A, C, G and T. According to their chemical composition the bases A and G, are called purines, and the bases T and C, are called pyrimidines. The genetic code is the set of rules by which information encoded in the DNA is translated into proteins. Specifically, the code defines a mapping between DNA-trimers called codons and amino acids. Every codon specifies a single amino acid. Amino acids are best known as the building blocks of proteins, and there are some 20 kinds of amino acids in proteins. The essential amino acids are those that the body does not synthesize in significant quantities, they are required in the diet. Plants produce these amino acids. The essential amino acids generally require complex biosynthetic pathways which humans do not have.

In order to complete the intensity table, four additional variables are added. Three of these variables are from the chemical context. The first variable is related to the composition of the DNA-dimer, it is to say, if the bases in the DNA-dimer are purines, pyrimidines or a combination. The second variable represents the number of amino acids mapped by a certain DNA-dimer, according to the genetic code. The third variable defines if the amino acids mapped by a DNA-dimer are essential or not. Finally the fourth variable is from the taxonomical context, it refers to the Genus of the Retroviral genome to which the DNA-dimer belongs. In figure 2, the main components of the target dataset generation step are shown.

## 4 Bayesian Networks Induction

In this section, the Bayesian Networks induction technique is described. We have selected the search based algorithms approach. The algorithms under this approach, have two components: a scoring metric and a search strategy. In this case we use the minimum description length principle as scoring metric, and a Genetic Algorithm as search strategy.

### 4.1 Bayesian Networks

Bayesian Networks are one of the best known formalisms to reason under uncertainty in Artificial Intelligence [7]. A Bayesian Network is a probabilistic graphical model that represents a given problem through a directed acyclic graph, in which the nodes represent variables, and the arcs encode the conditional dependencies between these variables. The arcs in the network can be thought of as representing direct causal relationships. Inducing a Bayesian Network, is the problem of finding a network that

best matches a training set of data. Finding a network means finding the structure of the graph and the conditional probability tables associated with each node. In the context of learning Bayesian Networks from data, there are two kinds of algorithms. Constraint based algorithms try to find out conditional dependencies between the variables, implicit in the data, in order to induce the structure. Search based algorithms perform a search in the space of possible network structures, looking for the structure that best fits the data, this search is guided by a scoring metric. Different measures to score competing networks as well as a variety of search strategies have been proposed.

## 4.2 The Minimum Description Length Principle

The minimum description length principle MDL, use the concept of conditional entropy related to the structure of a Bayesian Network [8]. The entropy is a non negative measure of uncertainty, maximal when total uncertainty is present, and zero under complete knowledge. When information is increased, the entropy decreases. This means that adding arcs to the network reduces entropy, because the probability distribution is better described. However, using exclusively entropy to guide the search, introduces a bias in favor of more complex networks, it is to say, densely connected networks. So, a property penalizing complexity is needed. The MDL has two terms, one measuring entropy, and another controlling complexity. Networks with small values for this metric are preferred.

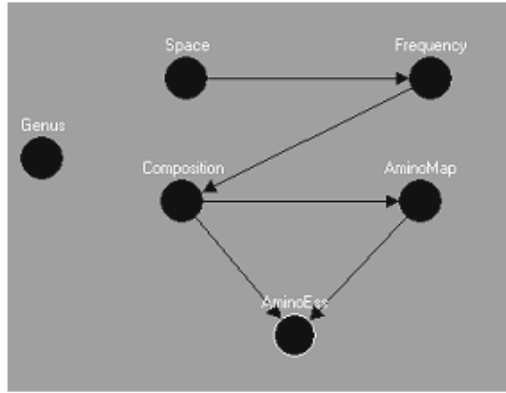
## 4.3 Genetic Search

The approach adopted to search the space of possible network structures, is to use a Genetic Algorithm. Genetic Algorithms are one of the well known paradigms of Evolutionary Computing [9]. The main idea of these paradigms is to simulate biological processes for complex function optimization. Genetic Algorithms are inspired in natural selection and genetics, operating on a population of individuals called chromosomes. Individuals are vectors in the hyperspace being explored. They represent possible solutions to the problem. In our approach, the solution corresponds to the structure of a Bayesian Network that minimizes the proposed metric. The structure of the network must be codified as an array of bits. The representation in our system, originally proposed by [10], codes the network using a connectivity matrix, where each row corresponds to a node in the network, and each column indicates with a one, if an arc from the node labeling the column to the node labeling the row exists. The absence of an arc is represented by zero. The connectivity matrix is transformed to a one dimensional array, concatenating the rows. The fitness function is the minimum description length principle described above. Standard genetic operators like selection, crossover and mutation, are applied in order to perform the optimization process. Generations are computed by rank selection, it means, sorting individuals by fitness with probability 0.5. Mutation is applied with probability 0.01. In figure 3, the main components of the Bayesian Networks induction strategy are shown.



$p(\text{Composition}=\text{Mix}|\text{Frequency}=\text{VLow})=0.93$   
 $p(\text{Composition}=\text{Pyr}|\text{Frequency}=\text{VHigh})=0.6$   
 $p(\text{Composition}=\text{Pur}|\text{Frequency}=\text{High})=0.58$   
 $p(\text{Frequency}=\text{VLow}|\text{Space}=\text{Mid})=1$   
 $p(\text{Frequency}=\text{VLow}|\text{Space}=\text{High})=1$   
 $p(\text{Frequency}=\text{Low}|\text{Space}=\text{Low})=0.95$   
 $p(\text{Frequency}=\text{Mid}|\text{Space}=\text{VLow})=0.54$

$p(\text{Space}=\text{VLow})=0.77$   
 $p(\text{Space}=\text{VHigh})=0.002$



$p(\text{AminoEss}=\text{Yes}|\text{AminoMap}=2, \text{Composition}=\text{Pyr})=1$   
 $p(\text{AminoEss}=\text{No}|\text{AminoMap}=1, \text{Composition}=\text{Pur})=1$   
 $p(\text{AminoEss}=\text{Mix}|\text{AminoMap}=3, \text{Composition}=\text{Mix})=1$   
 $p(\text{AminoMap}=1|\text{Composition}=\text{Pyr})=0.75$   
 $p(\text{AminoMap}=2|\text{Composition}=\text{Pur})=0.75$   
 $p(\text{AminoMap}=3|\text{Composition}=\text{Mix})=0.12$

**Fig. 4.** DNA-dimer Retroviral characterization

pyrimidines are related to a high Frequency, in contrast to those with a mixture composition which are highly related to a very low Frequency. In the case of the AminoMap variable, there is a clear differentiation, in the sense that there is a high probability that the DNA-dimers composed of pyrimidines map to one amino acid, the DNA-dimers composed of purines map to two amino acids, and those with a mixture composition map to three amino acids. The AminoEss variable has two causal relationships from the variables Composition and AminoMap. In this case there is a high probability that the amino acids be essential when the related DNA-dimer is composed of pyrimidines and map to two amino acids. Contrary to the non essential amino acids, which are related to a DNA-dimers composed of purines and map to one amino acid. By the time of this writing we are studying this DNA-dimer Retroviral characterization, in order to establish biological interpretations. We have applied this methodology to different taxonomical viral families, preliminary results show that each family has its own DNA-dimer distribution pattern. We have the certainty that the knowledge generated by these studies, will provide important directions in future Retroviral genomic investigations.

## 6 Conclusions

Pattern identification studies of DNA are of interest because the insight that they can provide about the evolutionary processes of species, the nature of life and the possible

biological and medical implications. A computational study of DNA-dimer distribution in Retroviral genomes based on a Bayesian Networks induction strategy was presented. We found interesting causal relationships between the variables studied, which have provided a characterization of the Retroviral genomes in the context of the DNA-dimer distribution. The computational methodology presented in this paper has demonstrated to be an interesting tool for the study and the analysis of genomic sequences. The analysis of the results for biological interpretations deserves a more detailed study.

## References

1. Quiroz-Gutierrez, A.: Biophysical Considerations and Evolutionary Aspects of DNA-dimer Frequency in AIDS Retrovirus Genomes. In: Topics in Contemporary Physics, pp. 239–248. IPN Press, México (2000)
2. Cocho, G., Medrano, L., Miramontes, P., Rius, J.L.: Selective Constrains over DNA Sequences. In: Biologically Inspired Physics. NATO ASI Series Physics, vol. 263, Plenum Press, New York (1991)
3. Breslauer, K.J., Frank, R., Blöcker, H., Marky, L.A.: Predicting DNA Duplex Stability from the Base Sequence. *Proc. Natl. Acad. Sci. USA* 83, 3746–3750 (1986)
4. Gladwin, M., Trattler, B.: *Clinical Microbiology*, Medmaster, Miami (2004)
5. van Regenmortel, M.H.V., et al.: *Virus Taxonomy, Classification and Nomenclature of Viruses*. In: Seventh Report International Committee on Taxonomy, Academic Press, USA (2000)
6. McNeill, D., Freiberger, P.: *Fuzzy Logic: The revolutionary computer technology that is changing our world*, Simon and Schuster, New York (1994)
7. Nilsson, N.J.: *Artificial Intelligence: a new synthesis*. Morgan Kaufmann, San Francisco (1998)
8. Boukaert, R.R.: Probabilistic network construction using the minimum description length principle, Technical Report, Utrecht University (1994)
9. Haupt, R.L., Haupt, S.E.: *Practical Genetic Algorithms*. John Wilen and Sons Inc, New York (1998)
10. Larrañaga, P.: Structure Learning of Bayesian Networks by Genetic Algorithms: A Performance Analysis of Control Parameters. *IEEE Journal on Pattern Analysis and Machine Intelligence* (1996)

# SELDI-TOF-MS Pattern Analysis for Cancer Detection as a Base for Diagnostic Software

Marcin Radlak<sup>1</sup> and Ryszard Klemous<sup>2</sup>

<sup>1</sup> School of Computer Science\*  
University of Birmingham

Edgbaston, Birmingham, B15 2TT

<sup>2</sup> Institute of Control and Optimization

Wroclaw University of Technology

ul. Wybrzee Wyspiaskiego 27

50-370 Wroclaw, Poland

msc63mpr@cs.bham.ac.uk or ryszard.klemous@pwr.wroc.pl

**Abstract.** The purpose of this paper is to present in an organized form the concept of cancer detection based on data obtained from SELDI-TOF-MS. In this paper, we outline the full process of detection: from raw data, through pre-processing towards classification. Methods and algorithms, their characteristics and suggested implementation indications are described. We aim to present the *state of the art* over current research. Additionally, we introduce an idea of 24h/day distributed work organization and suggest how to make the research process faster.

## 1 Introduction

There is no doubt that cancer is very serious disease which every year affects millions of people all over the world. To reduce this amount, extensive work, by research centers funded by pharmaceutical companies or charity organizations, is conducted to develop methods of prediction, prevention and treatment. The purpose of this paper is to introduce the concept of cancer detection using data obtained from SELDI-TOF-MS. This type of Mass Spectrometer has been proposed, because of its ability to produce high resolution spectrogram of proteins content in an organic sample. Assuming, that cancerous cells consist of proteins which are usually absent in healthy tissue, there is a hope to develop a method, which will able to distinguish between those two states, giving a solid base for final diagnosis which doctors have to make. Although, hardware is a breakthrough in the field it is still not precise enough to produce superior results expected from diagnosis equipment. Low repeatability of results, high noise and huge amount of data are only few of the difficulties encountered. Therefore, to supplement hardware deficiency, it is important to utilize more or less intelligent data analysis methods. Properly selected might, as some research show, significantly improve accuracy of the hardware. In this paper, we will outline

---

\* This paper has been funded by EPSRC.



the full process of detecting cancerous/healthy sample: from raw data, through pre-processing towards classification. Methods and algorithms, their characteristics and suggested implementation indications are presented to show what has been used and to encourage future work. They are collected in an organized way and aim to present the state of the art over current research. Additionally, we introduce an idea of 24h/day distributed work organization. This form of research and projects realization, which includes many participants that are not limited by geographical location or time, may provide an excellent opportunity to accelerate current work in this and many other fields. Finally, the summary and future directions are proposed to create some general indicators of the future work and to point a research in the correct direction.

The paper is organized as follows: in section 2 we present details about SELDI-TOF-MS technology. In section 3 full process of data manipulation: a) preprocessing, b) analysis and c) detection. In section 4 we provide details of 24h research process organization. Finally, we give a conclusion in section 6.

## 2 Proteomics

Mass spectrometry is based on a process of turning sample content into ions, isolating them and detecting according to the ratio between mass and charge. This general example shows many problems with this technology of which the most problematic one is ionization - how to ionize molecules to avoid splitting into smaller parts which could be detected as two smaller mass proteins. During the decade of the 1990s, changes in MS instrumentation and techniques revolutionized protein chemistry and fundamentally changed the analysis of proteins. These changes were catalyzed by two technical breakthroughs in the late 1980s: development of two ionization methods electrospray ionization (ESI) and matrix-assisted laser desorption/ionization (MALDI). But it was *Surface Enhanced Laser Desorption Ionization* method, which boosted the research in this field as previous methods lacked in consistency - each experiment, within the same conditions and settings, should produce the same results. By achieving this, comparison between samples was possible for further investigation of disease.

### 2.1 Sample Preparation for SELDI-TOF-MS

At first, an array has to be chosen, according to what type of proteins will be examined. Samples are applied and incubated on the spots of chosen array. Spot surfaces allow crystallization of specific proteins which are bounded to array. Any remaining unneeded material is washed out. Next, laser beam is applied to the spot causing desorption and ionization of the proteins. After this step, multiple spectra from statistically meaningful area are then averaged and a final spectrum is produced. Described steps are illustrated on figure [1](#).

Resulting mass spectrogram does not present masses of the proteins. It shows Mass to Charge ratio (M/Z) which is actually measured by Mass Spectroscop. Y axis represents intensity of ions at every M/Z ratio. This spectrogram provides

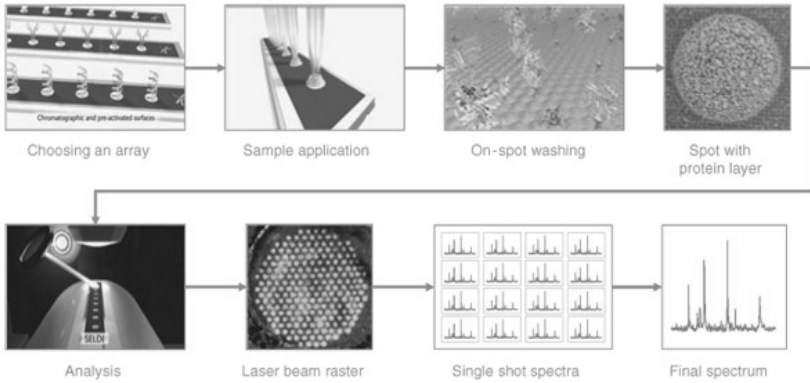


Fig. 1. Steps of sample processing [1]

an overview of protein content in sample. It is obvious that cancer cells contain many different proteins, therefore the goal is to detect a *pattern* of proteins in cancerous sample which is absent in healthy one. Following section presents methods which may be used to achieve this goal.

### 3 Sequence of Analysis

The idea is to classify produced spectrogram as cancerous or healthy. Figure 2 presents 216 samples of ovarian cancer grouped into cancer (black) and healthy (grey) - freely available from National Cancer Institute website <http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp>.

It is clear that spectrograms for cancer and normal samples differ significantly. From presented figure it is easy to classify samples, but often single spectrograms do not contain proteins which all remaining spectrograms of the same class

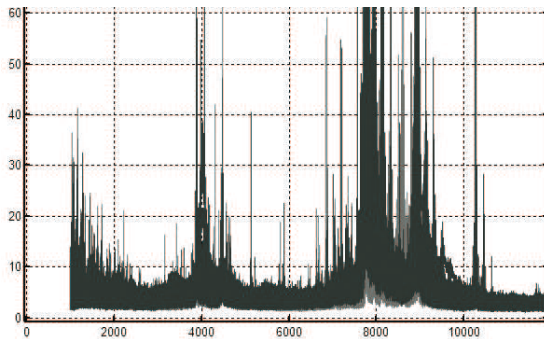


Fig. 2. Differences between samples: cancer(grey) and normal(black)

consists of. This may lead to misclassification. With medical software, any false classification is unacceptable.

Therefore, many issues arise when analyzing spectrograms. First of them is difficulty in detecting peaks - often high intensity for specific M/Z ratio in different samples is not high enough to be significant. There is also problem of overlapping samples, presence of noise, shifts in values (vertical and horizontal) and many more with high dimensionality at the end. To systemize process of data manipulation, following steps should be performed: preprocessing, analysis and classification. This way success - correct classification - is more likely to be achieved

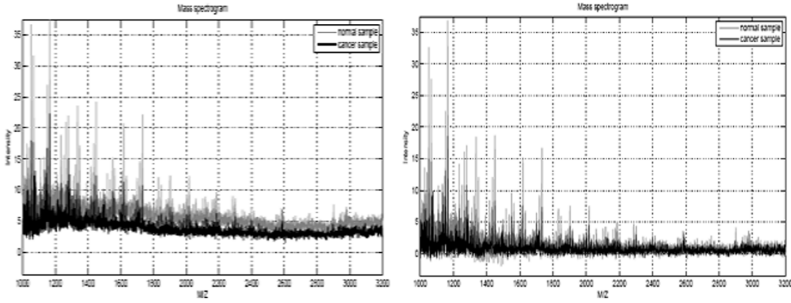
### 3.1 Preprocessing

Preprocessing can significantly increase performance of classifier. To be able to classify different samples, it is important to prepare them for this process. Different factors should be eliminated to prepare data for analysis and classification. If the samples were not preprocessed, classifiers may detect normal sample as cancer, or cancer as normal what, if it was medical tool, could result in very dramatic consequences. Person who is healthy could be mistakenly diagnosed as having cancer and would be referred for unnecessary treatment which is costly and very depressing. On the other side, ill person diagnosed as healthy, may lose chance of being cured, before the disease state is advanced. Following are some suggested preprocessing methods

**Dimension reduction.** Data used for the purpose of this paper is large and requires huge computational power. It consists of 216 samples of approx. 350 000 points (M/Z values) from the range of 1000mz to 12000mz. One file per sample is approx 5MB in size. Multiplying it by 216 (number of all samples) results in more than 1GB of memory required only to load the data. Many methods used in later analysis require duplication of the data so even more memory is required. Finally, operating on such a big data set is time consuming. Size reduction should allow elimination of insignificant values, leaving important ones for further analysis. At this stage, methods like PCA (Principal Component Analysis) performs very well.

**Baseline correction.** Samples are often shifted up and their minimum value is almost never 0 - Fig 3(left). It is common that for lower M/Z values, Intensity is much higher than for higher M/Z values. This may be avoided by aligning samples in a way presented on figure 3 - intensity begins around 0 and is not shifted up. On figure 3 cancer sample (grey) is shifted up according to normal sample (black). To eliminate maximum number of differences between samples, baseline correction has been applied. Intensities are shifted to 0 and their values were kept unchanged. This is a first step towards normalization of samples.

**Normalization.** Normalization is to re-scale intensity values to fit between defined range. This step is performed to emphasize differences between samples where they actually occur, rather than those which originate from different



**Fig. 3.** Baseline correction: before (left), after (right)

measurement conditions. One approach is to standardize the area under the curve to the group median. Afterwards, sample is standardized to be between specified range.

**Denosing.** As every digital data, mass spectrogram contains noise, which is impossible to avoid with every type of electronic device. There are many techniques for removing noise and many research has been done towards denoising this type of data. One idea is to use proposed undecimated discrete wavelet transform [2] as a tool to significantly improve efficiency and reproducibility of mass spectrometry. Other paper [3] suggest the use of denoising algorithm based on gaussian distributed noise. Having the mass spectrogram standardized and normalized, removing noise can will significantly improve the performance of classifier.

**Peaks alignment.** While baseline correction was used to shift samples vertically, it is also important to align them horizontally. There are different sources of shifts which may result in peak being present at different M/Z values. It could lead to misinterpretation and while peaks should occur at the same M/Z values, they could be detected as two different ones. Peaks alignment algorithm will be able to rescale data horizontally in a way, that every peak which is the same for every spectra is evenly placed.

After preprocessing is performed, analysis using PCA is a next step in data manipulation.

### 3.2 Data Analysis

Principle Component Analysis (PCA) is a mathematical transformation of a set of many correlated variables into a smaller set of uncorrelated variables - principal components. This is performed as follows ([4]):

1. Prepare data set
2. Subtract the mean
3. Calculate the covariance matrix

4. Calculate the eigenvectors and eigenvalues of covariance matrix
5. Choose components and form feature vector
6. Derive new data set

Most significant advantage of PCA is a smaller data set thus classifier will work faster.

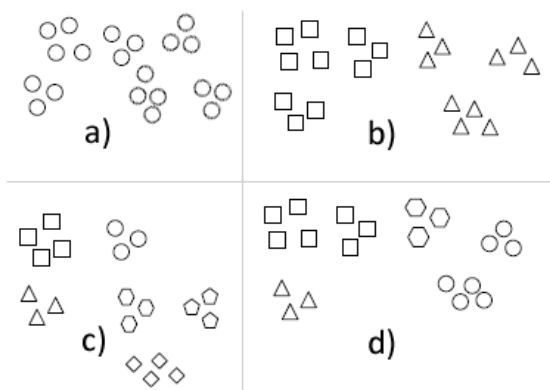
### 3.3 Classification

To develop a software which can be used for cancer diagnosis based on mass spectrometry, classifying algorithm is required. Previous steps were used to prepare raw data so noise and technology deficiencies were corrected. This ensures that input for classifier is always in the same range, with real rather than artificial differences allowing to distinguish between normal or cancer sample. With this requirement in mind many classifying algorithm have been developed and all the research work is focused on developing new ones, which will be applicable to the specific data set. Following, brief description of different approaches have been presented.

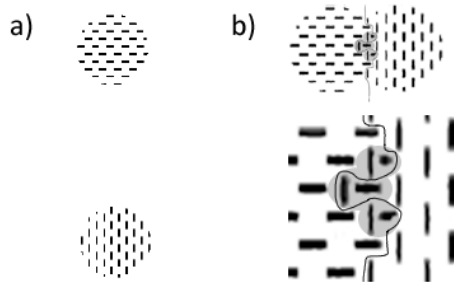
The main idea of classification is to be able to extract areas of similar properties and group them into classes. By creating those areas of similarities (particularly in unsupervised classification) it is possible to build a model which generalizes the properties of new data to be classified into one of the classes. Example of classification is presented on figure 4

Problem of correct classification arises when data are overlapping themselves. This is caused when data has similar parameters, so it can not be straightforward classified to one of the classes. Problem is presented on figure 5 That is why it is important to use a classifier, which will be able to distinguish differences and classify data correctly.

Classification algorithms can be divided into two major groups as proposed by Lilien et al. (2003). First, those algorithms which depend on the data and



**Fig. 4.** Classification example: a) input data (unclassified), b) two classes, c) six classes, d) four classes



**Fig. 5.** Classes: a) Well separated, b) overlapping

experiment conditions, return different results. The output is not deterministic, so they are called heuristic. Second group of algorithms are mathematical models, which for specific input, always return the same output. Those are called deterministic or exact models.

### Heuristic Approach to Classification

*Neural Networks.* Artificial Neural Networks (ANN) is a mathematical model of human brain. It is based on cooperation of many nonlinear processing units (neurons) connected in network. As real neurons, artificial ones (simplified) have inputs and outputs. By inputs they collect outputs from neurons they are connected to, and if the input is strong enough to activate the neuron, it gives the output. By combining neurons in multi layer networks, it is possible to solve many complex problems, which are impossible to solve (in reasonable time) by traditional mathematics i.e. they tend to approximate functions very well with reasonably small computing power requirements. Some of ANN's applications include: speech and pattern recognition, image recognition, financial prediction and many more. They also perform very well in proteomics and the results are described in following papers [5], [6] or [7]. Difficulty with Neural Network is that there are many parameters to set. It is thus very laborious task to adjust them correctly so the error on the output is minimized.

*Evolutionary Computations.* Genetic Algorithms are widely used in many fields, thus proteomics has not been omitted, i.e. [8], [9]. By applying evolutionary operators, i.e. crossover or mutation, search space is explored intelligently with successive generation. Each produced solution is then evaluated and compared to other in the population. Selection scheme defines which of them are chosen for next generation. Difficulty with this approach lies in fitness function definition.

**Deterministic Approach to classification.** While heuristic approach is sometimes the only way of solving problem, it is very common that results differ even if input data has not been changed. But sometimes it is possible to use exact algorithm, which will always produce determined result, for the same input data.

There are many tools available, but only some of them can be applied for mass spectrometry analysis. Few are described as follows.

*Linear Discriminant Analysis (LDA)*. Implemented in so called Q5 algorithm by [10], seems to perform very well when compared to other methods. The dimension has to be reduced before LDA is applied so (PCA) is performed to accomplish it. LDA, similar to PCA, searches for linear combination of variables which best describes the data. But the difference is that LDA also models differences between them whereas PCA does not explore it. As defined, "LDA approaches the problem by assuming that the probability density functions  $p(\mathbf{x}|y = 1)$  and  $p(\mathbf{x}|y = 0)$  are both normally distributed, with identical full-rank covariances  $\Sigma_{y=0} = \Sigma_{y=1} = \Sigma$ . It can be shown that the required probability  $p(y|\mathbf{x})$  depends only on the dot product  $\mathbf{w} \cdot \mathbf{x}$  where  $\mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$ . That is, the probability of an input  $\mathbf{x}$  being in a class  $y$  is purely a function of this linear combination of the known observations." When LDA classifier is trained, mean and covariance is calculated because those parameters are not known. But training in this case is shorter than time consuming heuristic methods.

*k-nearest neighbors*. k-Nearest Neighbors is another method of cluster analysis. The algorithm examine data and divide them into a predefined number of classes. Those classes contains categories of parameters which are derived from data during training process. After algorithm is trained, when a test sample is applied, classifier finds the k nearest neighbors and assigns their label names to the training data set. Importance of each neighbor is weighted by its rank presented in terms of the distance to test sample.

### 3.4 Verification

Every classifier has to be verified to show how well does it perform for input data. Data set is divided into 2 subsets: training set and verification set and it is important to perform this process properly. For freely available mass spectrometry data, quantity of samples is limited. Therefore k-fold cross validation, bagging or boosting algorithms have been proposed to increase the data set. To decide about classifier's efficiency, few terms have been proposed to describe how classifier performs. They are explained on figure 6:

PPV - sensitivity - is used to exclude disease presence

NPV - specificity - is used to confirm disease presence

$$sensitivity = \frac{true\ positive}{true\ positive + false\ negative}$$

$$specificity = \frac{true\ negative}{true\ negative + false\ positive}$$

Classifier performance may be described as a percentage of positive tests which correctly indicate the presence of disease (called positive predictive value - PPV),

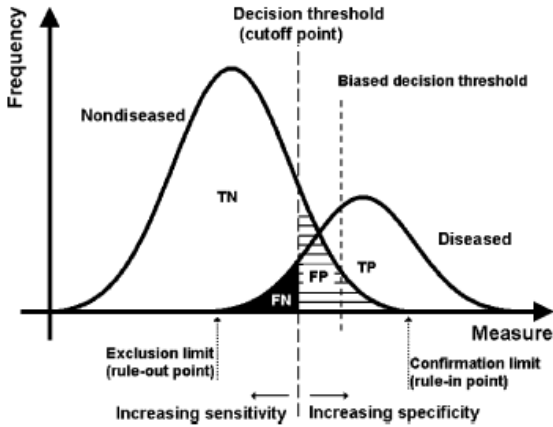


Fig. 6. Probability density of two class samples [11]

or percentage of negative tests which correctly indicate the absence of disease (negative predictive value - NPV). Further details about validation requirements were suggested by [11]

## 4 24h Work Organization - Accelerating Research

Based on [12] and [13] an idea of round the clock work organization could be introduced which might result in significant research acceleration. The idea is based on splitting production process in factory into 8 hours shifts. In this case, development of a product is limited by some area, where it is assembled. By utilizing time differences between different time zones, well developed communication and skilled experts living around the world - research could be done much faster, i.e. one research group based in UK could work 8 hours from 9am till 5pm. When their working day is finished, another research group based in time zone shifted 8 hours, could continue work on the same project. Afterwards, third research group could work when the previous has finished their working day. Greatest advantage of this approach is that each group can work within their best brain activity hours. The need for work over night is eliminated. This approach needs further development. We suggest 24h approach especially for the subject we described in this paper. It is very suitable because of modularity of data manipulation process: preprocessing, analysis, classification. Each module can be developed independently, thus there is no serial dependency. We would like to encourage research groups to develop relations with many other which could later be used in described way.



## 5 Conclusion

Whole decision process while diagnosing a disease is very complicated and requires a lot of knowledge and experience from person responsible for it. To make it easier, diagnosis methods are required to perform some tasks automatically, allowing larger group of less qualified specialists to be able to perform diagnosis. This paper outlined process of samples classification: steps which should be followed. More work is required in the field as it is very promising area of research. First, hardware requirements are still very high and demand for more accurate and repeatable machines is still huge. Data produced by apparatus have to be preprocessed in order to achieve scalability, and to adjust differences caused by noise or any other experimental obstacles and current methods should be improved or new developed. Furthermore, classifier is dependant on the data, thus deeper investigation is required to find proper methods for specific data set. This decision may be supported by statistical analysis of results. Finally, to make a process of developing new methods faster, we encourage researchers to focus some attention to possibilities which 24h work organization system provides.

## References

1. Vorderwülbecke, S., Cleverley, S., Weinberger, S.R., Wiesner, A.: Protein quantification by the seldi-tof-ms-based proteinchip system. *Nature Methods* 2 (2005)
2. Coombes, K.R., Tsavachidis, S., Morris, J.S., Baggerly, K.A., Hung, M.-C., Kuerer, H.M.: Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform. *Proteomics* 5(16) (2005)
3. Satten, G.A., Datta, S., Moura, H., Woolfitt, A.R., da Carvalho, M.G., Carlone, G.M., De Barun, K., Pavlopoulos, A., Barr1, J.R.: Standardization and denoising algorithms for mass spectra to classify whole-organism bacterial specimens. *Bioinformatics* 20(17) (2004)
4. Smith, L.I.: *A tutorial on Principal Components Analysis* (2002)
5. Ball, G., Mian, S., Holding, F., Allibone, R.O., Lowe, J., Ali, S., Li, G., McCardle, S., Ellis, I.O., Creaser, C., Rees, R.C.: An integrated approach utilizing artificial neural networks and seldi mass spectrometry for the classification of human tumours and rapid identification of potential biomarkers. *Bioinformatics* 18(3) (2002)
6. Zhou, Z.-H., IEEE, S.M., Liu, X.-Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering* (2005)
7. Yu, J., Chen, X.-W.: Bayesian neural network approaches to ovarian cancer identification from high-resolution mass spectrometry data. *Bioinformatics* 1 (2005)
8. Jeffries, N.O.: Performance of a genetic algorithm for mass spectrometry proteomics. *BMC Bioinformatics* (2004)
9. Boisson, J.-C., Jourdan, L., Talbi, E.-G., Rolando, C.: Protein sequencing with an adaptive genetic algorithm from tandem mass spectrometry. *Evolutionary Computation* (2006)
10. Lilien, R.H., Farid, H., Donald, B.R.: Probabilistic disease classification of expression-dependent proteomic data from mass spectrometry of human serum. *Journal of Computational Biology* 10(6) (2003)

11. Vitzthum, F., Behrens, F., Anderson, A.N., Shaw, J.H.: Proteomics: From basic research to diagnostic application a review of requirements and needs. *Journal of Proteome* 4 (2005)
12. Chaczko, Z., Klempous, R., Nikodem, J., Rozenblit, J.: 24/7 software development in virtual student exchange groups: Redefining the work and study week. In: *ITHET 7th Annual International Conference*, Sydney, Australia (2006)
13. Gupta, A., Seshasai, S., Arun, R.: Toward the 24-hour knowledge factory – a prognosis of practice and a call for concerted research (2006)

# Three Dimensional Modeling of Individual Vessels Based on Matching of Adaptive Control Points

Na-Young Lee

Korea Atomic Energy Research Institute, P. O. Box 105, Yuseong, Daejeon, Korea, 305-353  
nayoung@kaeri.re.kr

**Abstract.** In this paper, we propose a method for 3D (three-dimensional) modeling of individual vessels based on matching of adaptive control points to help accurately locate a disease such as arteriosclerosis. The proposed method consists of two steps: matching of corresponding control points between standard and individual vessels model, and transformation of standard vessels model. In the first step, control points are adaptively interpolated in the corresponding standard vessels image in proportion to the distance ratio if there were control points between two corner points in an individual vessels model. And then, the control points of corresponding individual vessels model matches with those of standard vessels model. In the second step, the TPS (Thin Plate Spline) interpolation function is used to modify the standard into the individual vessels model. In the experiments, we used patient angiograms from the coronary angiography in Sanggye Paik Hospital.

**Keywords:** Angiography, vessels, control point, image transformation, matching, interpolation.

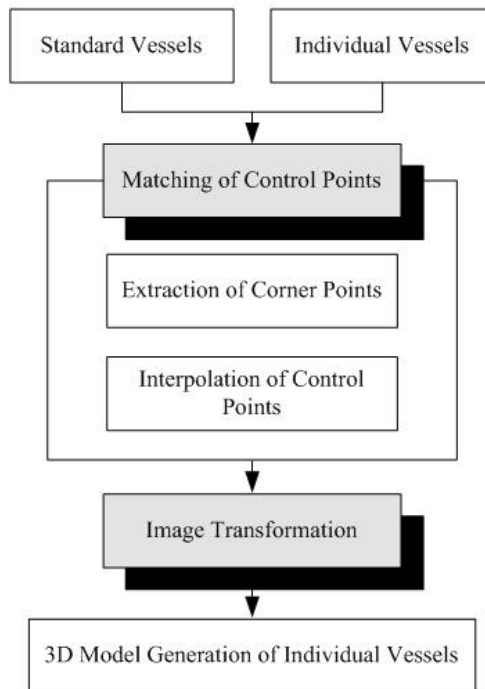
## 1 Introduction

Coronary artery diseases are usually revealed using angiographies. Such images are complex to analyze because they provide a 2D (two-dimensional) projection of a 3D object. Medical diagnosis suffers from inter- and intra-clinician variability. Therefore, reliable software for the 3D modeling of the coronary tree is strongly desired. Angiograms have been used clinically for diagnosing acute heart diseases such as atherosclerosis. Due to the ambiguity arising from a single projected view of a 3D elongated structure, a single plane angiogram alone is unreliable.

3D model provides many important anatomical measurements that are neither available, nor can be accurately measured in 2D. For example, the projected length of vessels is shorter in the projected views. Torque and the curvature of vessels are virtually impossible to estimate from 2D views. Accurate diagnosis can only be provided with sufficient knowledge and understanding of the anatomy. There are difficulties, especially when interpreting a 2D image of an organ with a 3D structure with various forms of curvature. Blood vessels in the human body are such organs. When a physician tries to insert a stent to treat a patient with aorta dissection or an artery disease, 3D images provide much easier and more accurate means of determining the exact location and size of the stent insertion. The 3D model provides

better and cleaner visualization allowing patient without extensive training to understand vessels geometry. It saves reviewing time for physicians since 3D model may be performed by a trained technician, and may also help visualize dynamics of the vessels.

Researchers in the 3D graphics field have been experimenting with new ways to more accurately visualize medical data-sets for the last two decades. As a result, two techniques have emerged which look promising. The first is advancement in the method of interpreting data-sets by generating a set of polygons that represent the anatomical surface, and displaying a 3D model representation. Polygons representing the outer surface of an object can be calculated using a variant of a “marching cubes” algorithm. The method of identifying surfaces of interest, referred to as segmentation, is generally a difficult problem for medical images. The second area of development, volume rendering, is a more direct way for reconstruction of 3D structures. Volume rendering represents 3D objects as a collecting of cube-like building blocks called voxels, or volume elements. Each voxel is a sample of the original volume, a 3D pixel on a regular 3D grid or raster. Each voxel has associated with it one or more values quantifying some measured or calculated property of the original object, such as transparency, luminosity, density, flow velocity or metabolic activity. The main advantage of this type of rendering is its ability to preserve the integrity of the original data throughout the visualization process. This technique, however, requires huge amounts of computation time and is generally more expensive than conventional surface rendering technique.



**Fig. 1.** Overall system configuration

In this paper, we propose a new approach for generating individual vessels model by modifying the standard vessels to suit the individual vessels model.

Overall system configuration is as shown in Fig. 1.

The structure of the paper is as follows. In Section 2 and Section 3, we describe the two major stages of our algorithm. Experimental results obtained for clinical datasets are discussed in Section 4. Finally, we discuss conclusion in Section 5.

## 2 Matching of Control Points

The shape of the arteries is quite different among individuals. For example, the lumen diameter is largely influenced by the length of the vessels, its width, and its tortuosity. It also depends on the sex, age, and pathological state of the individual. All these considerations contribute to the difficulty of building a coronary model.

Therefore, we automatic generation of 3D vessels model based on standard vessels model [2]. To transform a standard vessels image into an individual vessels image, it is important to match corresponding control points of the two images. In this paper, we used the feature points of vessels images as control points to automatically extract control points. Feature points here refer to the corner points of an object or points with higher variance of brightness compared to surrounding pixels in an image, which are differentiated from other points in the image. Such feature points can be defined in many different ways. They are sometimes defined as points that have a high gradient in different directions, or as points that have properties that do not change in spite of specific transformations. Grossly feature points can be divided into three categories. The first one uses non-linear filter, such as SUSAN corner detector proposed by Smith which relates each pixel to an area centered by the pixel. In this area, which is called SUSAN area, all pixels have similar intensities as the center pixel. If the center pixel is a feature point (some times a feature point is also referred to as a "corner"), its SUSAN area is the smallest one among the pixels around it. SUSAN corner detector can suppress noise effectively for it does not need the derivative of image. The second one is based on curvature, such as Kitchen and Rosenfeld's method. This kind of method needs to extract edges in advance, and then find out the feature points using the information of curvature of edges. The disadvantage of that kind of methods is that they need complicate computation, e.g. curve fitting, thus their speeds are relatively slow. The third kind of method exploits the change of pixel intensity. The typical one is Harris and Stephens' method. It produces corner response through eigenvalues analysis. Since it does not need to use slide window explicitly, its speed is very fast. Accordingly, this paper used the Harris corner detector to find feature points. However, the Harris corner detector has the problem of mistaking non-corner points with a high eigenvalue as corner points. To solve this problem of the Harris corner detector, we extracted control points and then found corner points among these control points. For accurate matching, the control points were adaptively interpolated in the corresponding standard vessels image in proportion to the distance ratio if there were control points between two corner points in an individual vessels model.

### 2.1 Extraction of Corner Points

In this paper, we performed thinning by using the structural characteristics of vessels to find the corner points among the feature points extracted with the Harris corner detector [2, 3].

A vascular tree can be divided into a set of elementary components, or primitives, which are the vascular segments, and bifurcation. Using this intuitive representation, it is natural to describe the coronary tree by a graph structure.

A vascular tree is consists of three vertices ( $v_{point}$ ) and one bifurcation ( $bif$ ).

These vertices are defined  $I_{thin}$  using the following equation (1).

$$I_{thin} = \{ v_{point}, bif \} \tag{1}$$

$$v_{point} = \{ v_{start\_point}, v_{end\_point1}, v_{end\_point2} \}$$

If the reference point is a vertex, the two control points closest to the vertex are called the corner points. If the reference point is a bifurcation, the three control points that are closest to it after comparing the distances between the bifurcation and all control points are called the corner points. As shown in Fig. 2, if the reference point is the vertex ( $v_{start\_point}$ ),  $v_1, v_2$  become the corner points; if the reference point is the bifurcation ( $bif$ ),  $v_3, v_6, v_9$  become the corner points.

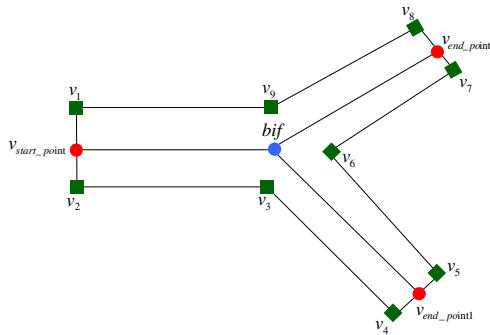


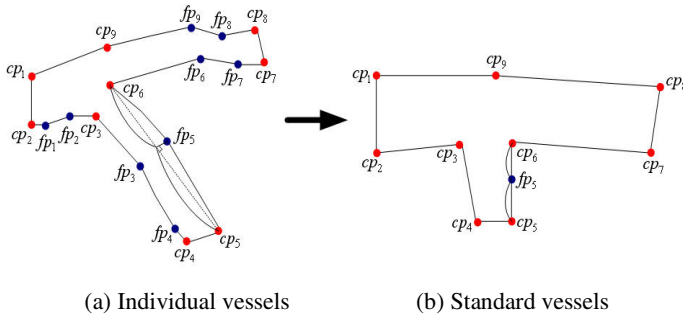
Fig. 2. Primitives of a vascular net

### 2.2 Interpolation of Control Points

For accurate matching, the control points were adaptively interpolated in the corresponding standard vessels image in proportion to the distance ratio if there were control points between two corner points in an individual vessels model.

The Fig. 3 shows interpolation of control points. The Fig. 3(a) shows that extracted control points from individual vessels image, and (b) shows an example that control point interpolated between standard vessels image and corresponding corner points

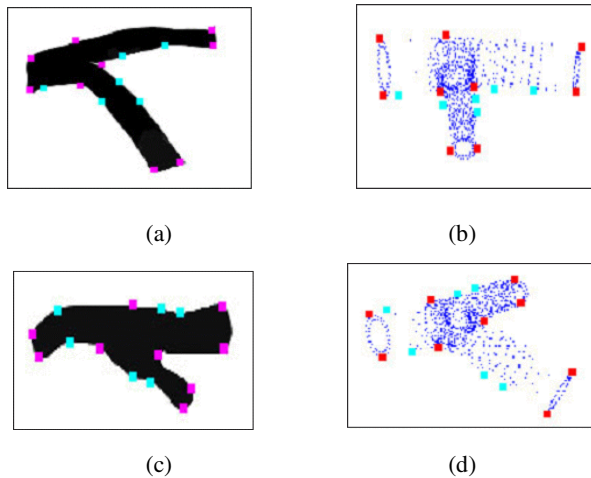
from (a) image. Where,  $fp$  are control points and  $cp$  are corner points which extracted from control points.



**Fig. 3.** Interpolation of control points

Control points of standard vessels model are interpolated by the distance rate between control point ( $fp_5$ ) and two corner points ( $cp_5, cp_6$ ) of individual vessels model.

The Fig. 4 shows the result of extracting the control points by applying the Harris corner detector to the vessels extracted from the angiogram.



**Fig. 4.** Result of adaptive corresponding interpolation of control points

### 3 Image Transformation

We have warped the standard vessels with respect to the individual vessels. Given the two sets of corresponding control points,  $S = \{s_1, s_2, \dots, s_m\}$  and  $I = \{i_1, i_2, \dots, i_m\}$ , 3D

model of vessels is performed by warping of the standard vessels. Here,  $S$  is a set of control points in standard vessels and  $I$  is a set of one in individual vessels.

In this work, standard vessels warping is performed by applying the elastic TPS(Thin-Plate-Spline) interpolation function[5] on the two sets of feature points.

The TPS are interpolating functions, representing the distortion at each feature point exactly, and defining a minimum curvature surface between control points. A TPS function is a flexible transformation that allows rotation, translation, scaling, and skewing. It also allows lines to bend according to the TPS model. Therefore, a large number of deformations can be characterized by the TPS model.

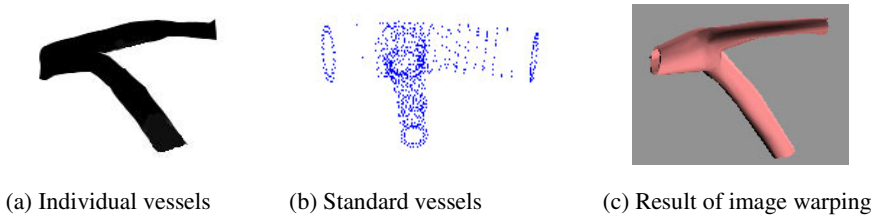
The TPS interpolation function can be written as

$$h(x) = Ax + t + \sum_{i=1}^m W_i K(\|x - x_i\|) \quad (2)$$

Where  $A$  are  $t$  are the affine transformation parameters matrices,  $W_i$  are the weights of the non-linear radial interpolation function  $K$ , and  $x_i$  are the control points. The function  $K(r)$  is the solution of the biharmonic equation ( $\Delta^2 K = 0$ ) that satisfies the condition of bending energy minimization, namely  $K(r) = r^2 \log(r^2)$ .

The complete set of parameters, defining the interpolating registration transformation is then used to transform the standard vessels. It should be noted that in order to be able to carry out the warping of the standard vessels with respect to the individual vessels, it is required to have a complete description of the TPS interpolation function.

The Fig. 5 shows the results of modifying the standard vessels to suit the model of individual vessels.



**Fig. 5.** Result of image transformation

## 4 Experimental Results

We simulated the system environment that is Microsoft Windows XP on a PentiumIV 3GHz, Intel Corp. and the compiler used was VC++ 6.0. The image used for experimentation was  $512 \times 512$ . Each image has a gray-value resolution of 8 bits, i.e., 256 gray levels.

The Fig. 6 shows the 3D model of standard vessels from six different angiographic views.



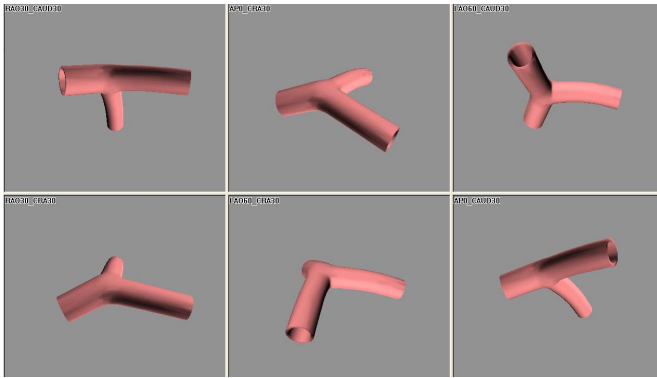


Fig. 6. 3D model of standard vessels in six views

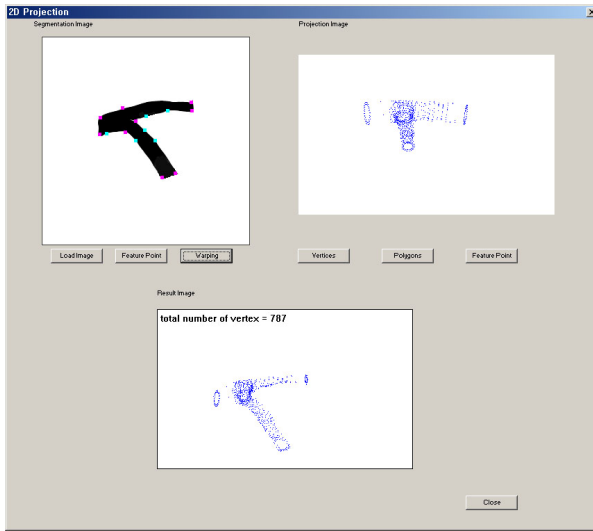


Fig. 7. Result of modifying of standard vessels

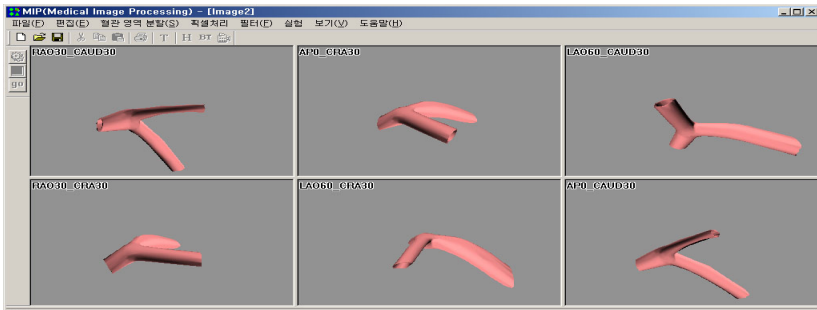


Fig. 8. Generation of individual vessels to 3D model in six views

The Fig. 7 shows the results of modifying the standard vessels to suit the model of individual vessels using the TPS interpolation function.

The Fig. 8 shows the result of automatic generating a 3D model of individualized vessels

## 5 Conclusion

We have developed a fully automatic algorithm to perform the 3D model of individual vessels from six angiograms, taken from a single rotational acquisition. As demonstrated in the previous section, this approach can be used to recover the geometry of the main arteries. The 3D model of vessels enables patients to visualize their progress and improvement. Such a model should not only enhance the level of reliability but also provides speedy and accurate identification. In order words, this method can expect to reduce the number of misdiagnosed cases.

## References

1. Lorenz, C., Renisch, S., Schlatholter, S., Bulow, T.: SPIE, Simultaneous Segmentation and Tree Reconstruction of the Coronary Arteries in MSCT Images. In: Proc. SPIE Int. Symposium Medical Imaging, vol. 5032, pp. 167–177 (2003)
2. Lee, N.Y., Kim, G.Y., Choi, H.I.: Automatic Generation Technique of Three-Dimensional Model Corresponding to Individual Vessels. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3984, pp. 441–449. Springer, Heidelberg (2006)
3. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: Proceedings of the Fourth Alvey Vision Conference, Manchester, pp. 147–151 (1988)
4. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on CVPR Seattle, pp. 593–600 (1994)
5. Bentoutou, Y., et al.: An invariant approach for image registration in digital subtraction angiography. *Pattern Recognition*, 34–48 (2002)
6. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. *International Journal of Computer Vision*, 151–172 (2000)
7. Brown, B.G., Bolson, E., Frimer, M., Dodge, H.: Quantitative coronary arteriography estimation of dimensions, hemodynamic resistance, and atheroma mass of coronary artery lesions using the arteriogram and digital computation. *Circulation* 55, 329–337 (1977)
8. de Feyter, P., Vos, J., Reiber, J., Serruys, P.: Value and limitations of quantitative coronary angiography to assess progression and regression of coronary atherosclerosis. *Advances in Quantitative Coronary Arteriography*, 255–271 (1993)
9. Ross, J., et al.: Guidelines for coronary angiography. *Circulation* 76, 963A-977A (1987)
10. Bookstein, F.L.: Principal warps: thin-plate splines and the decomposition of deformations. *IEEE-PAMI* 11, 567–585 (1989)
11. Flusser, J., Suk, T.: Degraded image analysis: an invariant approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 590–603 (1998)
12. Venot, A., Lebruchec, J.F., Roucaÿrol, J.C.: A new class of similarity measures for robust image registration. *Comput. Vision Graph. Image Process.* 176–184 (1998)

# Design and Implementation of Petri net Based Distributed Control Architecture for Robotic Manufacturing Systems

Gen'ichi Yasuda

Department of Human and Computer Intelligence, Faculty of Informatics  
Nagasaki Institute of Applied Science  
536 Aba-machi, Nagasaki 851-0193, Japan  
YASUDA\_Genichi@NiAS.ac.jp

**Abstract.** In this paper, the methods of the modeling and decomposition of the large and complex discrete event manufacturing systems are considered, and a methodology is presented for hierarchical and distributed control, where the cooperation of each controller is implemented so that the behavior of the overall system is not deteriorated and the task specification is completely satisfied. First, the task specification is defined as a Petri net model at the conceptual level, and then transformed to the detailed Petri net representation of manufacturing processes. Finally, the overall Petri net is decomposed and the constituent subnets are assigned to the machine controllers. The machine controllers are coordinated so that the decomposed transitions fire at the same time. System coordination through communication between the coordinator and machine controllers, is presented. Modeling and control of large and complex manufacturing systems can be performed consistently using Petri nets.

**Keywords:** Distributed control, industrial robotics, manufacturing systems, Petri nets.

## 1 Introduction

To implement large and complex discrete event manufacturing systems, it is necessary to provide effective tools for process modeling and sensing and machine control algorithms development. Large and complex systems have operation features such as parallelism and concurrency. The increasing use of robots and other computer-controlled machines has generated control software written in different levels and/or forms. It is required for users to understand all the control specifications without prejudice to the system's flexibility and maintainability. One of the effective methods to describe and control such systems is the Petri net which is a modeling tool for asynchronous and concurrent discrete event systems. Conventional Petri net based control systems were implemented based on an overall system model. Since in the large and complex systems, the controllers are geographically distributed according to their physical (hardware) structure, it is desirable to realize the hierarchical and distributed control.

The overall structure of the working area in a large and complex manufacturing system consists of one or more lines, each line consists of one or more stations, and each station (shop or cell) consists of one or more machines such as robots and intelligent machine tools. Inside of a cell, machines execute cooperation tasks such as machining, assembling and storing. Inside of a shop, cells cooperate mutually and execute more complicated tasks. The manufacturing system handles complicated tasks by dividing a task hierarchically in this structure, which is expected to be effective in managing cooperation tasks executed by great many machines or robots.

The hierarchical and distributed control for large and complex discrete event manufacturing systems has not been implemented so far. If it can be realized by Petrinets, the modeling, simulation and control of large and complex discrete event manufacturing systems can be consistently realized by Petrinets. In this paper, the author presents a methodology by extended Petrinets for hierarchical and distributed control of large and complex robotic manufacturing systems, to construct the control system where the cooperation of each controller is implemented so that the behavior of the overall system is not deteriorated and the task specification is completely satisfied.

## 2 Discrete Event Process Control Using Petrinets

A manufacturing process is characterized by the flow of workpieces or parts, which pass in ordered form through subsystems and receive appropriate operations. Each subsystem executes manufacturing operations, that is, physical transformations such as machining, assembling, or transfer operations such as loading and unloading. From the viewpoint of discrete event process control, an overall manufacturing process can be decomposed into a set of distinct activities (or events) and conditions mutually interrelated in a complex form. An activity is a single operation of a manufacturing process executed by a subsystem. A condition is a state in the process such as machine operation mode.

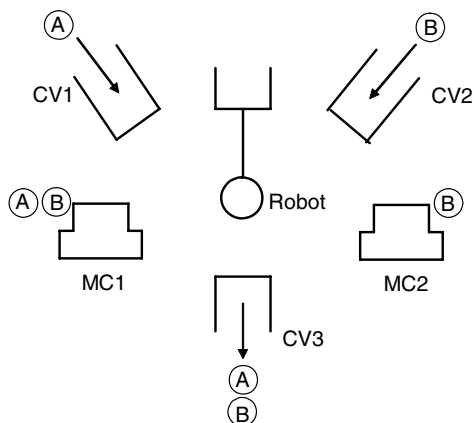
To represent discrete event manufacturing systems a modeling technique was derived from Petrinets [1]. The guarantee of safeness and the additional capability of input/output signals from/to the machine are considered. The execution rule for a transition is defined by the following clauses:

- (1) a transition is firable if one token is present in each of its input places and if there are not other firable transitions with higher priority;
- (2) if a transition is firable then it will eventually fire or become unfirable;
- (3) when a transition fires, one token is removed from each of the input places, one token is added to each of output places, and the action associated with the transition is executed.

By the representation of the activity contents and control strategies in detail, features of discrete event manufacturing systems such as ordering, parallelism, asynchronism, concurrency and conflict can be concretely described through the extended Petrinet.

### 3 Modeling and Decomposition Procedure

In this section the basic procedures of modeling and decomposition of robotic manufacturing systems are shown through a simple example. The specification of an example robotic manufacturing system shown in Fig. 1 is as follows.



**Fig. 1.** Example of Robotic Manufacturing System

- (1) The conveyor CV1 carries a workpiece of type A into the workcell.
- (2) The conveyor CV2 carries a workpiece of type B into the workcell.
- (3) The robot loads the workpiece of type A into the machining tool MC1.
- (4) The robot loads the workpiece of type B into the machining tool MC1 or MC2.
- (5) The machining tools process the workpieces each.
- (6) The robot unloads the processed workpieces and carries them to the conveyor CV3.
- (7) The conveyor CV3 carries the workpiece away.

A global, conceptual Petrinet model is first chosen which describes the aggregate manufacturing process. At the conceptual level the manufacturing process is represented using the Petrinet as shown in Fig. 2, where the activities of the resources such as robots, conveyors and other machines, are also represented. Based on the hierarchical approach, Petrinets are translated into detailed subnets by stepwise refinements from the highest system control level to the lowest machine control level [2]. At each step of detailed specification, some parts of the Petrinet, transitions or places, are substituted by a subnet in a manner, which maintains the structural properties. Fig. 3 shows the detailed Petrinet representation for loading, processing and unloading in Fig. 2.

It is natural to implement a hierarchical and distributed control system, where one controller is allocated to each control layer or block. For the manufacturing system, an example structure of hierarchical and distributed control is composed of one station

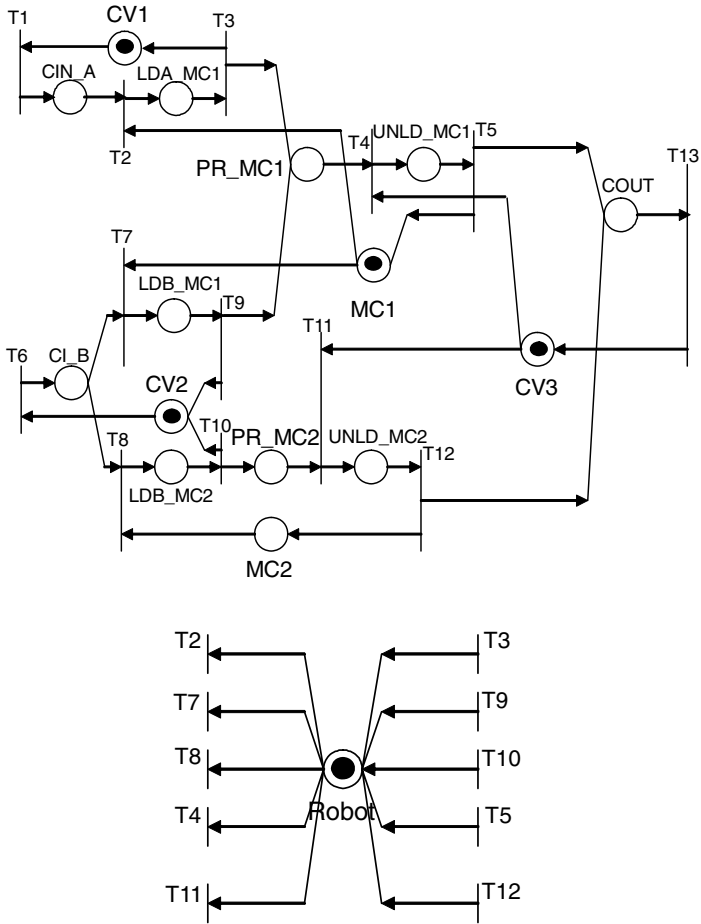
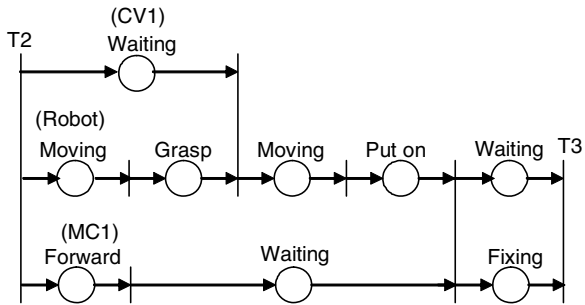
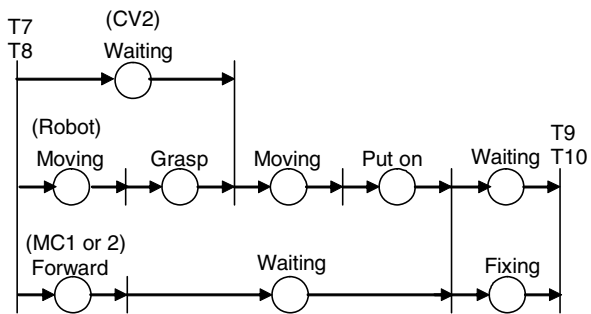


Fig. 2. Petri net representation of the example system at the conceptual level

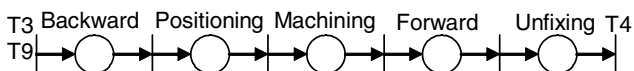


(a) Loading A to MC1 (LDA\_MC1)

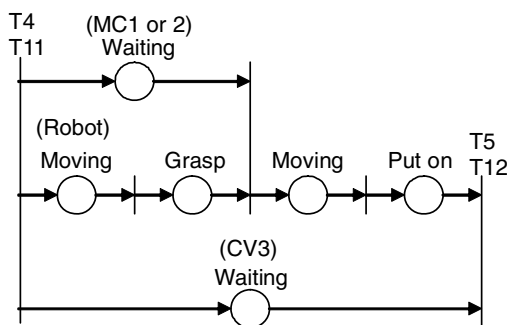
Fig. 3. Detailed Petri net representation



(b) Loading B into MC1 or MC2 (LDB\_MC1, LDB\_MC2)



(c) Processing (PR\_MC1, PR\_MC2)



(d) Unloading A or B (UNLD\_MC1, UNLD\_MC2)

Fig. 3. (continued)

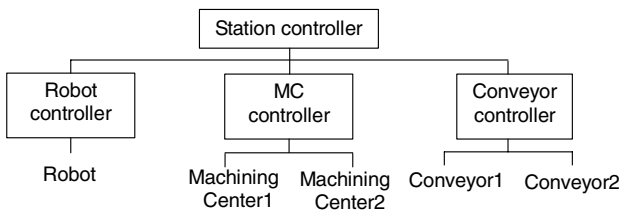


Fig. 4. Example structure of distributed control system

controller and three machine controllers (Fig. 4), although each robot may be controlled by one robot controller. The detailed Petrinet is decomposed into subnets, which are executed by each machine controller. In this step, a transition may be divided and distributed into different machine controllers. The machine controllers should be coordinated so that these transitions fire in union. The Petrinet executed in each machine controller is shown in Fig. 5.

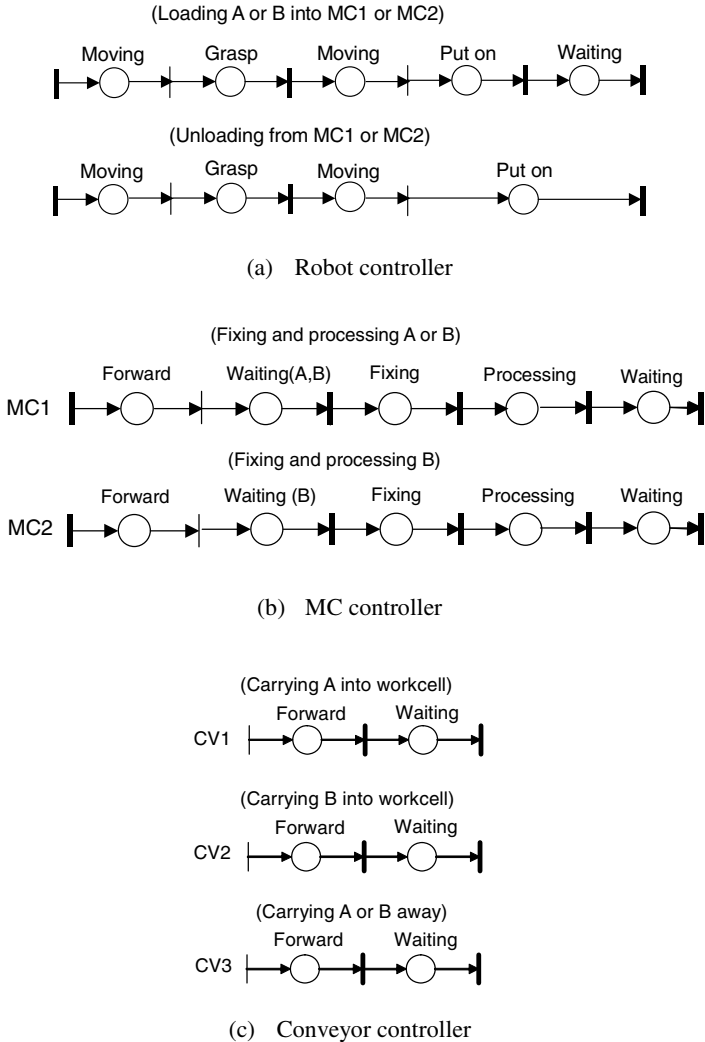


Fig. 5. Petrinet representation of machine controllers



### 4 Coordination of Transitions

The station controller must coordinate the machine controllers so that the decomposed transitions fire in union. Such transitions are called global transitions, and other transitions are called local transitions [3], which are shown in Fig. 6. It is proved that the firability condition of the original transition is equal to AND operation of firability conditions of decomposed transitions. In case that a transition in conflict with other transitions is decomposed as shown in Fig. 7, these transitions should be coordinated by the station controller.

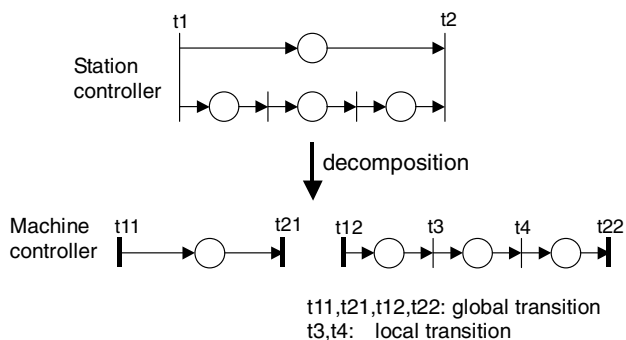


Fig. 6. Decomposition of transition

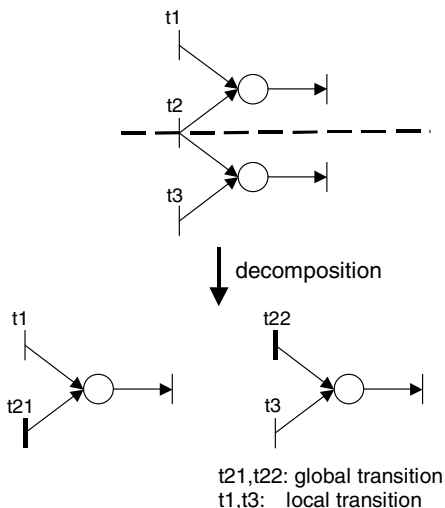
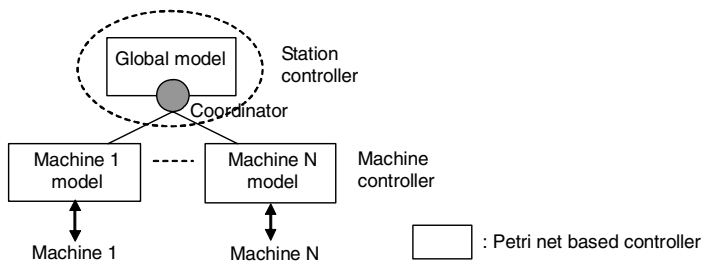


Fig. 7. Decomposition of transition in conflict

The control software is distributed into the station controller and machine controllers. The station controller is composed of the Petrinet based controller and the coordinator. The conceptual Petrinet model is allocated to the Petrinet based controller for management of the overall system. The detailed Petrinet models are allocated to the Petrinet based controllers in the machine controllers. Each machine controller directly monitors and controls the sensors and actuators of its machine. The control of the overall system is achieved by coordinating these Petrinet based controllers. The control structure of a two-level control system is described in Fig. 8. System coordination is performed through communication between the coordinator in the station controller and the Petrinet based controllers in the machine controllers as the following steps.

- (1) When each machine controller receives the start signal from the coordinator, it tests the firability of all transitions in its own Petrinet, and sends the information on the global transitions and the end signal to the coordinator.
- (2) The coordinator tests the firability of the global transitions, arbitrates conflicts among global and local transitions, and sends the names of firing global transitions and the end signal to the machine controllers.
- (3) Each machine controller arbitrates conflicts among local transitions using the information from the coordinator, generates a new marking, and sends the end signal to the coordinator.
- (4) When the coordinator receives the end signal from all the machine controllers, it sends the output command to the machine controllers.
- (5) Each machine controller outputs the control signals to its actuators.



**Fig. 8.** Structure of two-level hierarchical and distributed control

Multilevel hierarchical and distributed control for large and complex manufacturing systems can be constructed such that the control system structure corresponds to the hierarchical and distributed structure of the general manufacturing system. The overall system is consistently controlled, such that a coordinator in a layer coordinates one-level lower Petrinet based controllers and is coordinated by the one-level upper coordinator. The coordination mechanism is implemented in each layer repeatedly. The details of coordination in hierarchical and distributed control composed of a global controller and several local controllers have been implemented as shown in Fig. 9. A conceptual view of the multilevel hierarchical and distributed control for large and complex manufacturing systems is shown in Fig. 10.

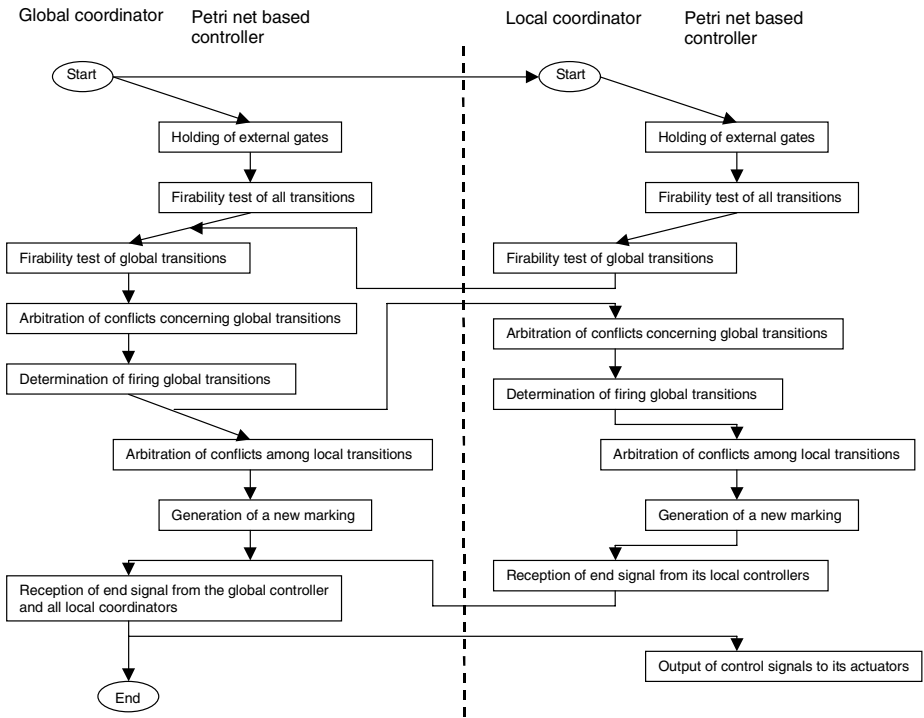


Fig. 9. Flowchart of coordination in hierarchical and distributed control

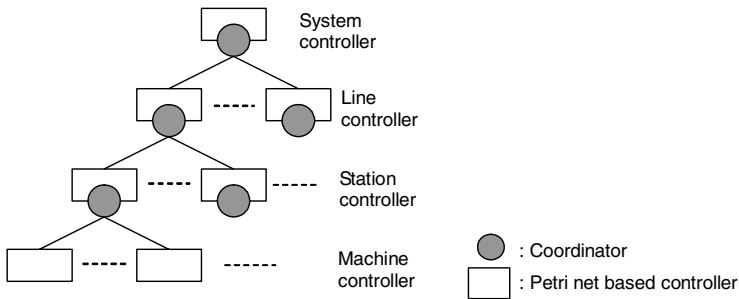
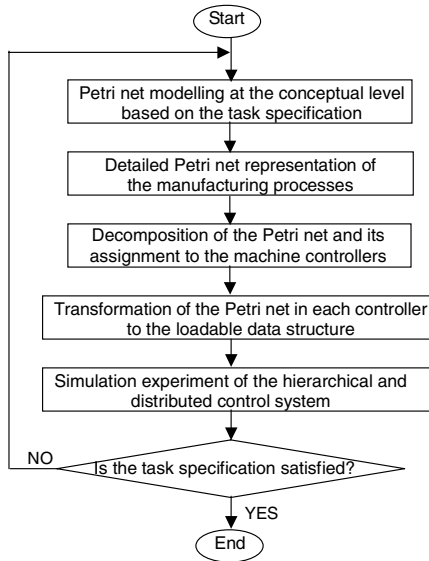


Fig. 10. Conceptual view of multilevel control system

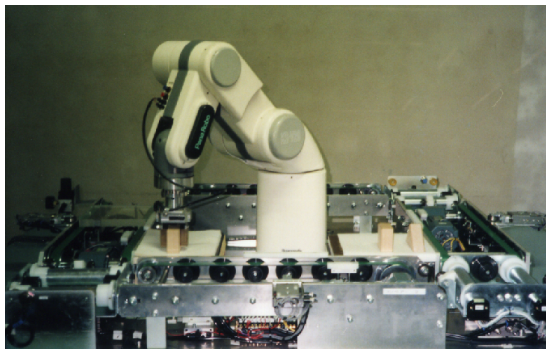
## 5 Implementation of the Control System

The overall procedure for the design and implementation of hierarchical and distributed control is summarized as shown in Fig. 11. For the example system, an experimental robot system has been constructed as shown in Fig. 12, and the hierarchical and distributed control system has been realized using a set of PCs.

Each machine controller is implemented on a dedicated PC. The robot controller executes robot motion control through the transmission of command and position data. The station controller is implemented on another PC. Communications among the controllers are performed using serial communication interfaces. A Petrinet model includes control algorithms, and is used to control the manufacturing process by coincidence of the behavior of the real system with the Petrinet model. A machine controller controls one or more machines or robots using multithreaded programming [4]. For cooperative or exclusive tasks between robots, global transitions are used to communicate the status of the robots.



**Fig. 11.** Flow chart of Petrinet based implementation of hierarchical and distributed control



**Fig. 12.** View of experimental robot system

The names of global transitions and their conflict relations are loaded into the coordinator in the station controller. The connection structure of a decomposed Petrinet model and conflict relations among local transitions are loaded into the Petrinet based controller in a machine controller. In the connection structure, a transition of a Petrinet model is defined using the names of its input places and output places; for example,  $t1-1=b1-1$ ,  $b1-1t1$ , where the transition no.1 ( $t1-1$ ) of the subsystem no.1 is connected to the input place no.1 and the output place no.1. Using the names of local transitions, global transitions are defined; for example,  $T1=t1-1$ ,  $t2-1$  indicates that the global transition  $T1$  is composed of the transition  $t1-1$  of the subsystem no.1 and the transition  $t2-1$  of the subsystem no.2.

## 6 Conclusions

A methodology to construct hierarchical and distributed control systems, which correspond to the structure of manufacturing systems, has been presented. The overall control structure of an example robotic manufacturing system was implemented using a communication network of PCs, where each machine controller is realized on a dedicated PC. The control software is distributed into the station controller and machine controllers; the station controller executes the conceptual Petrinet, and the machine controllers execute decomposed subnets. By introduction of the coordinator, the controllers are arranged according to the hierarchical and distributed nature of the manufacturing system. The Petrinet model in each machine controller is not so large and easily manageable. Modeling, simulation and control of large and complex manufacturing systems can be performed consistently using Petrinets. The experimental control system uses conventional PCs with serial interfaces, but the performance of the control system can be improved using dual port RAM or high-speed serial interfaces for communication between controllers.

## References

1. Hasegawa, K., Takahashi, K., Miyagi, P.E.: Application of the Mark Flow Graph to Represent Discrete Event Production Systems and System Control. *Trans. of SICE* 24, 69–75 (1988)
2. Yasuda, G., Tachibana, K.: Implementation of Communicating Sequential Processes for Distributed Robot System Architectures. In: *IFAC Manufacturing Systems: Modelling, Management and Control 1997*, pp. 321–326. Pergamon Press, Oxford (1997)
3. Yasuda, G.: Hierarchical and Distributed Control of Discrete Event Robotic Manufacturing Processes by Extended Petri Nets. In: *Proceedings of the 4th Asia-Pacific Conference on Industrial Engineering and Management Systems (APIEMS 2002)* (2002)
4. Yasuda, G.: Distributed Control of Multiple Cooperating Robot Agents using Multithreaded Programming. In: *Proceedings of the 16th International Conference on Production Research* (2001)

# Multi Sensor Data Fusion for High Speed Machining

Antonio Jr. Vallejo<sup>1</sup>, Ruben Morales-Menendez<sup>2</sup>,  
Miguel Ramírez<sup>1</sup>, J.R. Alique<sup>1</sup>, and Luis E. Garza<sup>2</sup>

<sup>1</sup> Instituto de Automática Industrial  
Carretera Campo Real km 0.200 La Poveda  
28,500 Arganda del Rey Madrid, Spain  
{avallejo, mramirez, jralique}@iai.csic.es

<sup>2</sup> Tecnológico de Monterrey, campus Monterrey  
Avenida Eugenio Garza Sada 2501  
64,849 Monterrey NL, México  
{rmm,legarza}@itesm.mx

**Abstract.** Surface roughness ( $Ra$ ) control in High Speed Machining ( $HSM$ ) demands reliable monitoring systems. A new data fusion model based on a multi-sensor system is developed. The model considers cutting parameters, cutting tool geometry, material properties and process variables. It can be used to predict the  $Ra$  *pre* and *in*-process. The Response Surface Design methodology was used to minimize the number of experiments. Artificial neural networks were exploited as data fusion techniques. Early results represent the building blocks for a low cost supervisory control system that optimizes the  $Ra$  in  $HSM$ .

## 1 Introduction

High performance machining implies the use of high technology for improving two aspects: high removal material rates and machining accuracy, specially the surface roughness ( $Ra$ ).

Boothroyd and Knight [1] consider that the final  $Ra$  during a practical machining operation is defined as the sum of two independent effects: the *ideal*  $Ra$ , due to the geometry of the tool and the feed rate; the *natural*  $Ra$ , due to the irregularities in the cutting operation. The *ideal*  $Ra$  represents the best possible finish that may be obtained. However, it is not possible to achieve in practice, and normally the *natural*  $Ra$  forms a large proportion of the actual roughness.

$Ra$  can not be measured online with high reliability because sensors are currently too inaccurate and too unreliable for effective use in many machining applications. Direct sensors are an impractical solution because of vibration and chip loads. Non-contacting sensors have interference from the environment. Indirect sensing such as *sensor fusion* are a practical solution. Several methodologies have been developed for *prediction* and *monitoring*  $Ra$  in different machining processes.

*Sensor fusion* is a mathematical method that integrates several sensor signals into one fused measurement. These integrated measurements can predict relevant states such as  $Ra$ , more accurately than single sensor measurements. Artificial neural networks and statistical multiple regression are options for this approach. With the available process information, the *sensor fusion* technique must be applied in complementary manner to provide a more robust prediction of the  $Ra$ .

The rest of the paper is organized as follows. Section 2 reviews relevant research in this field. The experimental setup, the data acquisition system and signal processing are defined in section 3. A key step in this research corresponds to the design of experiments, which is included in section 4. The sensor fusion models are reported in section 5. The results are discussed in section 6. Finally, section 7 concludes the paper.

## 2 State of the Art

$Ra$  has been an important design feature and quality measure in many machined parts; so, it has been investigated for many years. Benardos and Vosniakos [2] consider four major approaches: machining theory, experimental research, design of experiments, and artificial intelligence.

*Machining Theory Approach.* This approach follows the theory of machining that considers process kinematics, cutting tool properties, chip formation mechanism, etc. Two modeling methods are considered: geometric and analytical.

Lee et al. [3] propose a geometric model for modeling the end mill offset and tilt angle. A simulation algorithm and programming method was used to simulate the machined surface. The  $Ra$  was computed by considering cutting parameters, cutter and workpiece geometry, and run-out parameters. The vibration was also included in order to improve the  $Ra$  prediction. An analytical model was introduced in [4]. An experimental design validated the model. The model was used for optimization of the  $Ra$  in up-face milling.

*Experimental Approach.* Regression analysis is employed in order to build models based on the experimental data sets.

Based on a regression model, [5] found that  $Ra$  is a quadratic function of tool service time. Also, the major parameters affecting  $Ra$  are the tool geometry and tool wear.

In [6], a correlation between  $Ra$  and vibration signals in turning is given. It is found that models with cutting parameters and tool vibrations are more accurate than those depending only on cutting parameters.

*Design of Experiments Approach.* This approach constitute a systematic method concerning the planing of experiments, collection and analysis of data with near-optimum use of available resources. The Response Surface Methodology (*RSM*) and Taguchi techniques for Design Of Experiments (*DOE*) are the most widespread methodologies for the  $Ra$  prediction problem.

A statistical model is presented in [7], where the  $Ra$  is estimated in high-speed flat end milling. A rotatable central composite design was considered with the  $RSM$  to compute a second order model. The most significant variables were the total machining time, depth of cut, step over, spindle speed and feed rate. Other works related with this methodology are described in [6,8].

*Artificial Intelligence approach.* Artificial Neural Network ( $ANN$ ), Genetic Algorithms ( $GA$ ), Fuzzy Logic ( $FL$ ) and expert systems are typical approaches used in the  $Ra$  prediction problem.

Azouzi and Guillot [9] combined the  $ANN$  modeling, statistical tool, and sensor fusion technique to estimate on-line  $Ra$  and dimensional deviations during the turning process. The design of experiments consider cutting parameters (feed, speed, and depth of cut), process conditions (coolant, tool wear, and material properties) and only one type of tool and workpiece material. The final  $ANN$  model was built using the feed and depth of cut plus the radial and feed forces from sensors.

A surface recognition system for  $Ra$  prediction in end milling process was proposed by [10]. An  $ANN$  model was built from the spindle speed, feed rate, depth of cut and the vibration average per revolution.

Benardos and Vosniakos [11] present an  $ANN$  model to recognize the  $Ra$  in a face milling process. Using the degree of freedom methodology, an  $ANN$  model is built using elements from the face milling theory. Also, [12] presented an online surface recognition system based on  $ANN$  using a sensing technique to monitor the effect of vibration produced by the cutting tool and workpiece during the turning process. Additional to the vibration signals, the spindle speed, feed rate and depth of cut are considered to develop the  $ANN$  model.

A multiple regression analysis for modelling the  $Ra$  was proposed by [13]. Three different materials were used and the model was built, after of an evaluation of sensory features, with forces and emission acoustic signals.

Even though previous works exhibit satisfactory results, more general and practical solutions are needed. Exploiting sensor fusion techniques could be a great opportunity. Sensor fusion modeling is a problem of signal estimation, interpolation and prediction. Fusion techniques can be classified into three different groups: (1) fusion based on probabilistic models such as Bayesian reasoning; (2) fusion based on least-squares techniques as Kalman filtering, optimal theory, regularization; (3) intelligent fusion that includes fuzzy logic, artificial neural networks, etc. The third group will be exploited in this paper.

### 3 Experimental Set Up

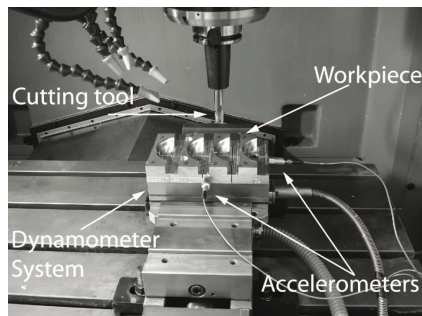
The experiments were conducted in a  $HSM$  center HS-1000 Kondia, with 25KW drive motor, three axis, maximum spindle speed 24,000  $rpm$ , and a Siemens open Sinumerik 840D controller (Figure 1). Several HSS end mill cutting tools ( $25^\circ$  helix angle, and 2-flute) from Sandvik Coromant were selected for the end milling process. Several workpiece materials (aluminum with hardness from 70 to 157 HBN ) were selected. These materials are used in the aeronautic and mold



manufacturing industry. Also, several cutting tool diameters (from 8 to 20 mm) were tested.

*Data Acquisition System.* Several sensors were installed in the *CNC* machine. For measuring the vibration, 2 PCB Piezotronics accelerometers were installed in x and y-axis directions. These instruments have a sensitivity of  $10\text{ mV/g}$ , in a frequency range from 0.35 to 20,000 *Hz*. Measurement range is  $\pm 500\text{ g}$ . The dynamic cutting force components ( $F_x, F_y, F_z$ ) were sensed with a 3 component force dynamometer, on which the workpiece was mounted.

All the signals were acquired with a high speed multifunction DAQ *NI-PCI* card, which ensures 16-bit accuracy, at a sampling rate of 1.25 *MS/sec*. The system was configured to acquire the signals with a sampling rate of 40,000 samples/sec.



**Fig. 1.** Experimental set up in the *CNC* milling center

*Signal processing.* Signals from the sensors must be processed to obtain relevant features that characterize  $R_a$ . Basically, the raw signals undergo three steps in the signal processing:

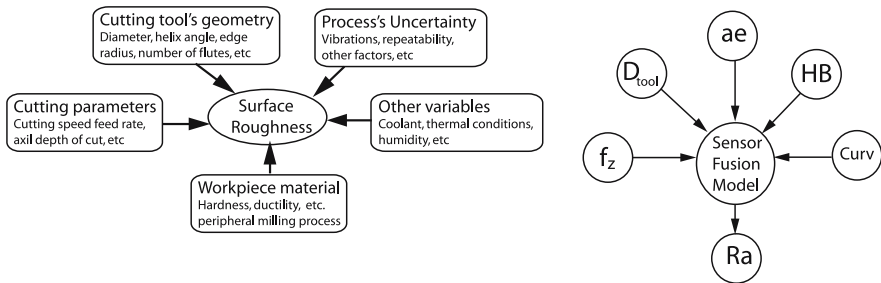
- (i) Signal segmentation. The signals were divided in small frames with 4,096 points each one. It corresponds at 0.1024 seconds of the machining time.
- (i) Features extraction. The statistical features (mean, root mean square, skew, kurtosis) were computed for all the frames of each signal.
- (iii) Average value. An average was computed of the feature based on the area where the  $R_a$  was measured (in the middle part of the machining surface).

## 4 Design of Experiments

The mechanics of material removal are complex because the  $R_a$  depends on several factors. These factors can be divided into different groups [2]. First group corresponds to the cutting parameters: cutting speed, feed rate, axial depth of cut, etc. The second group involves the geometry of the cutting tool: tool diameter, helix angle, edge radius, number of flutes, etc. The third group is

**Table 1.** Factors and levels. Important factors are: *feed per tooth* ( $f_z$ , mm/rev), *cutting tool diameter* ( $D_{tool}$ , mm), *radial depth of cut* ( $ae$ , mm), *hardness of the workpiece* ( $HB$ ,  $HBN$ ), and *curvature of the machining geometry* ( $Curv$ ,  $mm^{-1}$ ).

Levels	$f_z$	$D_{tool}$	$ae$	$HB$	$Curv$
-2	0.025	8	1	71	-0.05
-1	0.05	10	2	93	-0.025
0	0.075	12	3	110	0
1	0.1	16	4	136	0.025
2	0.13	20	5	157	0.05



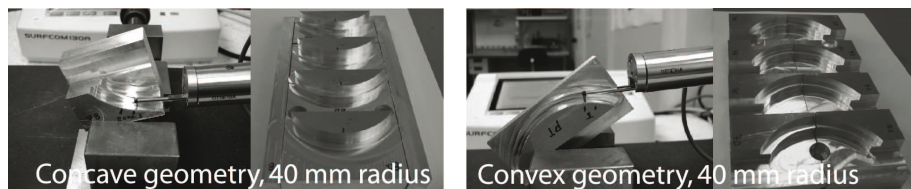
**Fig. 2.** Surface Roughness’s variables. Left side shows all factors that affect  $Ra$ , while right diagram isolates the most important ones.

defined by the workpiece material: hardness, ductility, etc, and geometry of the peripheral milling process: concave, convex or straight path. The last group results from the uncertainty of the process due to the variations in machine vibrations and repeatability, work-holding devices, and other factors. Variables less important are coolant, thermal conditions, humidity, and so on. Figure 2 shows these defined groups (left side).

It is very important to determine which factors represent the main effects over  $Ra$  during the machining process. The proposed factors and levels were defined by applying a screening factorial design with eight factors and two extreme levels.

After the screening phase, five factors are taken as the most influent on  $Ra$ . An optimization phase was designed in order to obtain a predictor model for the response. The design of experiment was a rotatable central composite design with  $\alpha = 2$  as the radius of the sphere and 16 points ( $2^{(k-1)} = 16$ ) on the cube, where  $k = 5$  represents the number of factors with 5 levels per factor. Also, 10 points outside of the cube and 6 central points were added.

A total of 128 experiments were done, 32 runs with 4 replicates. The main factors and levels appears in Table 1 and the right diagram in Figure 2. These levels are representatives of the common working range for each physical variable. The curvature factor represents the inverse of the radius of the workpiece, which is negative if convex and positive otherwise.



**Fig. 3.** Geometries and the measurement of the  $R_a$

The machined workpieces were  $100\text{ mm} \times 170\text{ mm} \times 25\text{ mm}$  blocks (Figure 3).  $R_a$  was measured using a portable roughness meter Surfcom 130A. Six measurements were taken over the machined surface.

## 5 Sensor Fusion Model

### 5.1 Statistical Regression

By applying an ANalysis Of VAriance (ANOVA) to the four replicates, it was possible to evaluate the effects due to the factor combinations ( $P < 0.05$ ), and identify the contribution of each factor using the statistical  $F$ . A polynomial equation was computed by using the factors more significant, and it is given by (1).

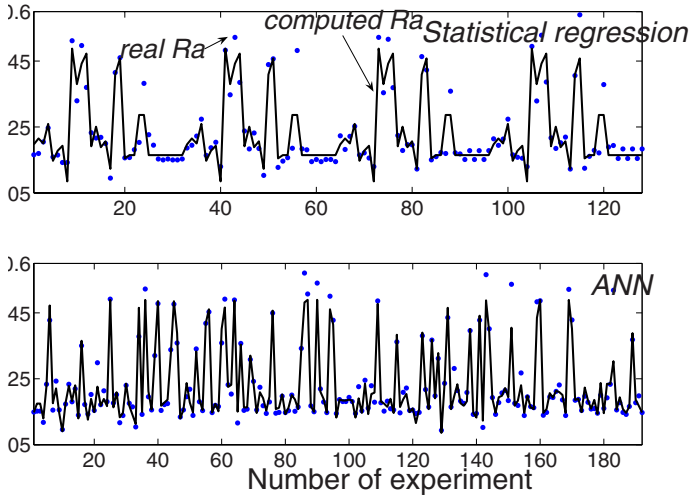
$$\begin{aligned}
 R_a = & 0.1646 + 0.0715f_z + 0.02536f_z^2 - 0.0758D_{tool} + 0.0354D_{tool}^2 + 0.0304HB^2 \\
 & - 0.0422f_zD_{tool} - 0.0224f_zHB - 0.0118D_{tool}(ae) + 0.0121D_{tool}HB \\
 & + 0.0193(ae)Curv
 \end{aligned} \quad (1)$$

This model has a squared error  $R^2 = 0.89$  and the adjusted squared error  $R_{adj}^2 = 0.88$ . Top plot in Figure 4 shows the experimental  $R_a$  versus the computed  $R_a$ . This model was validated by analyzing the histogram of raw residuals which follows a normal distribution. Also, the normal probability plot of the raw residuals follows a linear relationship. Finally, there is an excellent spread of points (no patterns) on either side of zero among raw residuals and computed values.

### 5.2 Artificial Neural Networks

The statistical model computes the  $R_a$  by considering the cutting parameters, cutting tool geometry and material properties, as show in the right diagram in Figure 2. This model predicts the  $R_a$  with minimum error when the information is within  $RSM$  domain. This model can be used to predict the  $R_a$  in *pre*-process machining; this is, before start machining operation. However, it is necessary to consider the *process variables* that allow to estimate the  $R_a$  value *in*-process machining.

An estimator based on multi-sensor and data fusion provides an improved and robust estimation. There are different frameworks for fusing signal features



**Fig. 4.** Comparison of the real  $Ra$  versus computed  $Ra$ . Top plot corresponds to the statistical regression model. Bottom plot represents the ANN performance.

such as mathematical functions, black-box models, rule-based fuzzy sets, ANN, etc. ANN framework has several attractive properties such as universal function approximation capabilities, unresponsive to noisy or missing data, and accommodation of multiple non-linear variables for unknown interactions.

Among the various ANN models, feed-forward architecture is a classic model, and back-propagation algorithm is an excellent training method for this structure. This ANN model has been used in several research works related to the machining process, [11], [14], and [10].

Based on a cross-correlation analysis exploiting the Pearson correlation index the RMS (Root Mean Square) acceleration in  $x$ -axis and  $y$ -axis ( $RMS Acc_x$  and  $Acc_y$ ) are the only variables considered due to the high correlation with  $Ra$ . Forces and other statistics (mean, skew, kurtosis) of the accelerations in both axis were discarded. However, if the  $x$  and  $y$  forces are fused as a total force,  $F_T = \sqrt{F_x^2 + F_y^2}$ ,  $F_T$  has important contribution to  $Ra$ . Finally, the new model for  $Ra$  based on ANN fuses cutting parameters ( $f_z$ ), cutting tool geometry ( $D_{tool}$ ) and material properties ( $HB$ ), and process variables ( $Acc_x$ ,  $Acc_y$ ,  $F_T$ ). The bottom plot in Figure 4 shows the performance of this model.

## 6 Results

For the ANN model, the experimental dataset were normalized to avoid numerical instability. Bipolar sigmoidal normalization was applied, because the minimum and maximum values are unknown in real-time. The non-linear transformation prevent that most values from being compressed into essentially the same values, and it also compress the large outlier values [14][15].

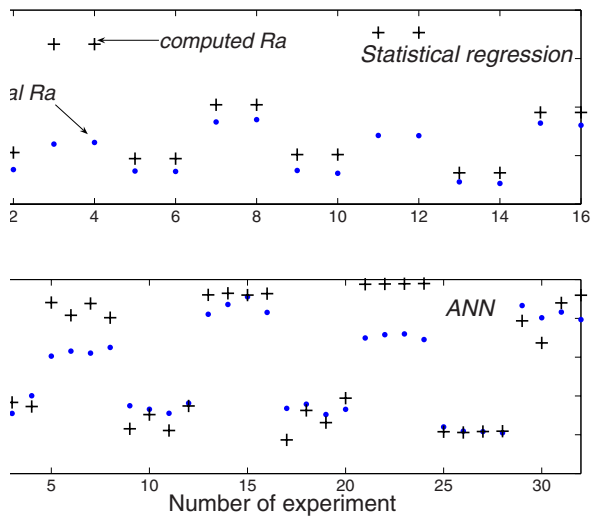
**Table 2.** Factors for testing the models

$f_z$	$D_{tool}$	$ae$	$HB$	$Curv$
0.04	12, 16	1, 5	69, 146, 150	-0.025, -0.05, 0.05, 0.025
0.013	12, 16	1, 5	69, 142	-0.025, -0.05, 0.05, 0.025

The statistical regression and ANN models were tested with a new dataset of experiments, which were defined according to Table 2.

### 6.1 Statistical Regression Approach

The model was trained and tested with different experimental datasets. The results are shown in the top plot of Figure 5. These results present a considerable deviation with respect to the actual values, with a regression R-value 0.698 and  $MSE = 0.0716$ .



**Fig. 5.** Comparison of real  $Ra$  versus computed  $Ra$ . Top plot shows the statistical regression’s performance when the model is tested with different experimental data; bottom plot shows same condition for ANN approach.

### 6.2 ANN Approach

Three ANN architectures were tested.  $ANN(n_i, n_{h1}, \dots, n_{hN}, n_o)$  represents a feed-forward network with  $n_i$  neurons in the input layer,  $n_{h1}$  neurons in the hidden layer #1,  $n_{hN}$  neurons in the hidden layer #N, and  $n_o$  neurons in the output layer.

**ANN(5,5,1).** This ANN architecture considers as inputs:  $f_z$ ,  $D_{tool}$ ,  $ae$ ,  $HB$  and  $Curv$ , one hidden layer with 5 neurons is considered and 1 output neuron for  $Ra$ . The ANN model was trained with a dataset of 3 replicates (96 data). The performance of the model is shown in the top plot of Figure 6. This model exhibits a regression R-value of 0.698 and a  $MSE = 0.0716$ . These results are similar at the corresponding statistical regression model.

**ANN(6,6,1).** This ANN architecture considers as inputs:  $f_z$ ,  $D_{tool}$ ,  $HB$ ,  $Acc_x$ ,  $Acc_y$ , and  $F_T$ , 1 hidden layer with 6 neurons is considered and 1 output neuron for  $Ra$  (middle plot of Figure 6). The model was trained with 96 data and testing with a new experimental database. The regression R-value was 0.889 and  $MSE = 0.0177$ .

**ANN(7,7,1).** This ANN architecture was configured with the cutting parameters:  $f_z$ ,  $D_{tool}$ ,  $ae$  and  $HB$ , and the process variables  $Acc_x$ ,  $Acc_y$ , and  $F_T$  to predict the  $Ra$ . This model implies the process variables should be considered. The number of training data points were of 192 conditions. The performance of the ANN model was better than other models, with a regression R-value 0.904 and  $MSE = 0.0056$ . Bottom of Figure 6 shows the results.

The improvement in reduction of the  $MSE$  was more a factor of 3 and 13 for ANN(5,5,1) and ANN(6,6,1) respectively.

With a good model for predicting  $Ra$ , a supervisory control system could be integrated as in Figure 7. Given a product specifications, the  $Ra$  model could be exploited *pre-process* in order to define the operating conditions. Then, by measuring and fused the process variables during the machining, a prediction of  $Ra$  can be obtained. Control actions could be taken if  $Ra$  prediction and

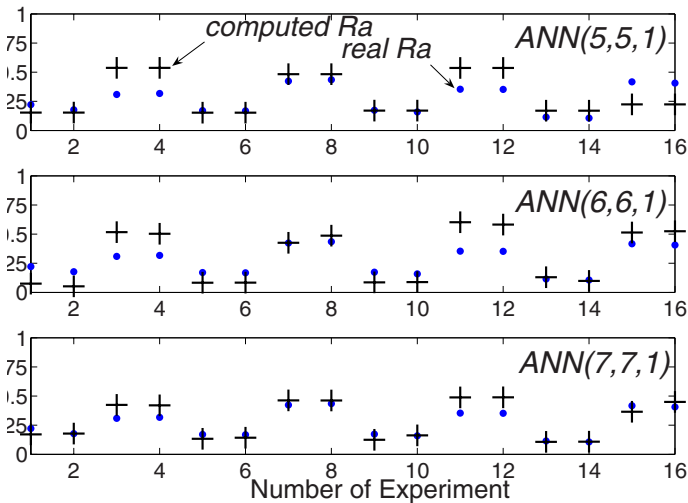


Fig. 6. Comparison of actual  $Ra$  versus computed  $Ra$  for different ANN models

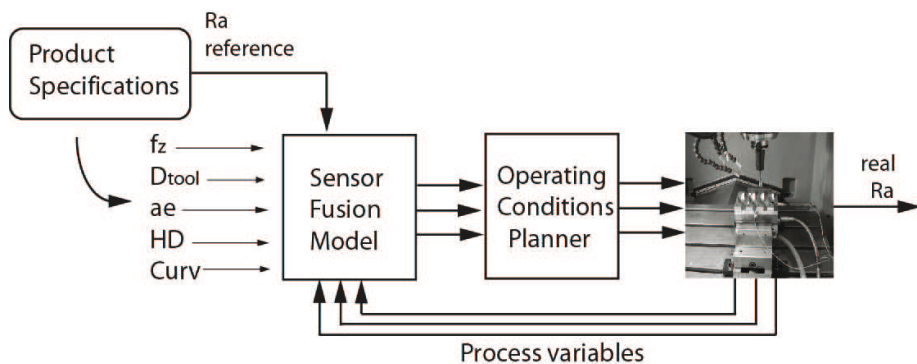


Fig. 7. Pre- and In- process planner

$Ra$  reference vary. This proposal represents the building blocks and inferential control system for  $Ra$  [16].

## 7 Conclusions

A statistical regression and several ANN models were tested for predicting  $Ra$  in high speed machining for peripheral milling processes. The main results were: (1) By exploiting the Surface Response Methodology, the number of experiments were reduced with excellent results, (2) the most important variables in this phenomena were identified, (3) the ANN based model showed good results in a more general range of variables, (4) sensor fusion increased the performance of the model prediction, and (5) the building blocks for a supervisory control system were identified.

## References

1. Boothroyd, G., Knight, W.A: Fundamentals of Machining and Machine Tools, 3rd edn. Taylor and Francis Group, Boca Raton (2006)
2. Benardos, P.G., Vosniakos, G.C.: Predicting surface roughness in machining: A review. Int. J. of Machine Tools and Manufacture, 833–844 (2003)
3. Lee, K.Y., Kang, M.C., Jeong, Y.H., Lee, D.W., Kim, J.S.: Simulation of surface roughness and profile in high-speed and milling. Materials Processing Technology (113), 410–415 (2001)
4. Sai, K., Bouzid, W.: Roughness modelling in up-face milling. Int J. Adv. Manuf. Technol. (26), 324–329 (2005)
5. Barber, G.C., Gu, R., Jiang, Q., Tung, S.: Surface roughness model for worn inserts of face milling: Part ii - an empirical model. Tribology Transactions 44(1), 142–146 (2001)
6. Abouelatta, O.B., Madl, J.: Surface roughness prediction based on cutting parameters and tool vibrations in turning operations. Materials Processing Technology (118), 269–277 (2001)

7. Ozcelik, B., Bayramoglu, M.: The statistical modeling of surface roughness in high-speed flat end milling. *International Journal of Machine Tools and Manufacture* 46, 1395–1402 (2006)
8. Ozel, T., Karpat, Y.: Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks. *Machine tools and Manufacture* (45), 467–479 (2005)
9. Azouzi, R., Guillot, M.: On-line prediction of surface finish and dimensional deviation in turning using neural network based sensor fusion. *Int. J. Mach. Tools Manufact.* 37(9), 1201–1217 (1997)
10. Tsai, Y.H., Chen, J.C., Lou, S.J.: An in-process Surface regression system based on neural networks in end milling cutting operations. *International Journal of Machine Tools and Manufacture* (39), 583–605 (1999)
11. Benardos, P.G., Vosniakos, G.C.: Prediction of surface roughness in cnc face milling using neural networks and taguchi's design experiments. *Robotics and Computer Integrated Manufacturing* (18), 343–354 (2002)
12. Lee, S.S., Chen, J.C.: On-line surface roughness recognition system using artificial neural networks system in turning operations. *International Journal of Advanced Manufacturing Technology* 22, 498–509 (2003)
13. Kwon, Y., Ertekin, Y.M., Tseng, T.: Identification of common sensory features for the control of cnc milling operations under varying cutting conditions. *International Journal of Machine Tools and Manufacture* 43, 897–904 (2003)
14. Feng, C.X., Wang, X.F.: Surface roughness predictive modeling: Neural networks versus regression. *IIE Transactions on Design and Manufacturing*, 1–42 (2002)
15. Sun, J., Hong, G.S., Rahman, M., Wong, Y.S.: Improved performance evaluation of tool condition identification by manufacturing loss consideration. *International Journal of Production Research* 43(6), 1185–1204 (2005)
16. Vallejo, A., Morales-Menendez, R., Alique, J.R.: Designing a cost-effective supervisory control system for machining processes. In: *IFAC-Cost Effective Automation in Networked Product Development and Manufacturing, Mexico* (to appear, 2007)



# VisualBlock-FIR for Fault Detection and Identification: Application to the DAMADICS Benchmark Problem

Antoni Escobet<sup>1</sup>, Àngela Nebot<sup>2</sup>, and François E. Cellier<sup>3</sup>

<sup>1</sup> Dept. ESAIL, Universitat Politècnica de Catalunya, Manresa, Spain  
toni@epsem.upc.edu

<sup>2</sup> Dept. LSI, Universitat Politècnica de Catalunya, Barcelona, Spain  
angela@lsi.upc.edu

<sup>3</sup> Institute of Computational Science, ETH Zurich, CH-8092 Zurich, Switzerland  
FCellier@Inf.ETHZ.CH

**Abstract.** This paper describes a fault diagnosis system (FDS) for non-linear plants based on fuzzy logic. The proposed scheme, named VisualBlock-FIR, runs under the Simulink framework and enables early fault detection and identification. During fault detection, the FDS should recognize that the plant behavior is abnormal, and therefore, that the plant is not working properly. During fault identification, the FDS should conclude which type of failure has occurred. The enveloping and acceptability measures introduced in VisualBlock-FIR enhance the robustness of the overall process. The final part of this research shows how the proposed approach is used for tackling faults of the DAMADICS benchmark.

## 1 Introduction

There has been an intensive research activity in the Fault Diagnosis System (FDS) area that includes quantitative as well as qualitative approaches. Quantitative approaches are primarily based on statistical techniques, first order logic, control theory, mathematical modeling, and computer simulation [1, 2, 3]. The main drawback of quantitative techniques is that they operate on a quantitative and precisely formulated plant model that is not always available. Also, human plant operators usually rely on heuristic knowledge that is easy to be captured by means of qualitative methodologies. There is a large amount of research done in the area of qualitative FDS, specially using expert systems, neural networks, and genetic programming [4, 5, 6, 7]. However in recent years, the demand has arisen to develop FDS that are more robust to uncertainty. In this context, fuzzy logic and hybrid fuzzy approaches appear to offer a good alternative to other qualitative FDS methodologies [8, 9, 10].

In this paper, a VisualBlock-FIR FDS based on fuzzy inductive reasoning (FIR) methodology is presented and applied to a pneumatic servomotor actuated control valve, which is the benchmark problem of the European research training network called DAMADICS. DAMADICS stands for Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems.

In the next section, the fault detection and identification processes of VisualBlock-FIR are introduced. Section 3 presents the case study and the results obtained. Finally, the conclusions of the research are presented in section 4.

## 2 Fault Detection and Identification Methodology

As mentioned before, a FDS needs to detect and identify the different faults that may occur in the system over time. VisualBlock-FIR performs these tasks by means of a user-friendly framework. The fault detection process of VisualBlock-FIR is described in Fig. 1.

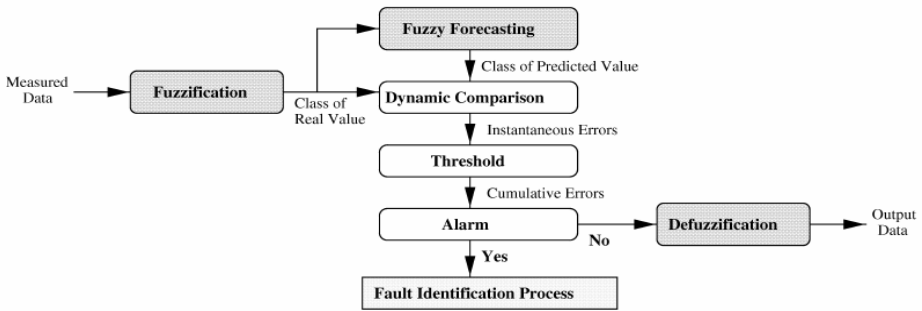


Fig. 1. VisualBlock-FIR Fault Detection Process

In Fig. 1 the grey boxes represent FIR processes, whereas the white boxes constitute the fault detection procedure. The data measured from the system is converted into qualitative triples (class, membership, and side) by means of the FIR fuzzification process. The fuzzy forecasting process predicts the next output value, a qualitative triple, from the qualitative data using the model (mask and pattern rule base) that represents the current behavior of the system. The fuzzy forecasting process computes also the enveloping interval that drives the detection process. The enveloping concept is based on the 5 nearest neighbors that are computed inside the FIR inference engine by means of the k-nearest neighbor rule. A value of 5 has been chosen because from a statistical point of view every state should be observed at least five times [11]. A distance measure is computed between the input pattern, from which the output prediction should be obtained, and all patterns stored in the pattern rule base that match with that input pattern. The 5 patterns with shortest distance are selected as the 5 nearest neighbors. For a deeper insight into the FIR methodology, the reader is referred to [12]. As shown in Fig. 2, the enveloping is composed by an upper bound (maximum value) and a lower bound (minimum value) delimiting the space where the real output signal should be present. It is important to notice that the enveloping

bounds are obtained directly from the 5 nearest neighbors, and therefore the predicted signal is always in the range set by the two bounds. This is not the case of the real signal. If a real value falls outside the envelope, an instantaneous error occurs, meaning that the model used in the prediction does not correctly represent the system in that specific point. The instantaneous errors occurred inside a predetermined time window are accumulated over time (see Fig. 2). When the cumulative errors within the window are greater than the threshold specified by the modeler, an alarm is issued, and it is then necessary to identify the fault that has occurred. A narrow enveloping interval implies that the 5 neighbors are very close to each other, meaning that the information available of the behavior of the system in that point is very rich and complete. By contrast, a wide enveloping interval means that there is not a lot of information about the system in that point, and therefore, the nearest neighbors are far away from each other. It is important to keep in mind that the FIR methodology is based on the system's behavior rather than on its structure, and therefore, the amount and richness of the data available from the system are crucial in order to assure the identification of an accurate and reliable model representing it.

Fig. 2 presents an example of FIR fault detection using the enveloping concept showing a time window of 15 prediction points. The upper and lower dotted lines represent the upper and lower bounds of the envelope, respectively, whereas the continuous line is the real output signal. In the bottom part of the figure the instantaneous errors are accumulated. As can be seen, the real value falls outside the envelope for the first time in point number 6 when the real value is bigger than the enveloping maximum value, causing an instantaneous error. The same occurs in points number 7 and 11. The threshold specified in this example was of 3 cumulative errors, and therefore, a fault alarm is triggered in point number 11 when the third instantaneous error arrives. The next step is the identification of the new fault that has occurred.

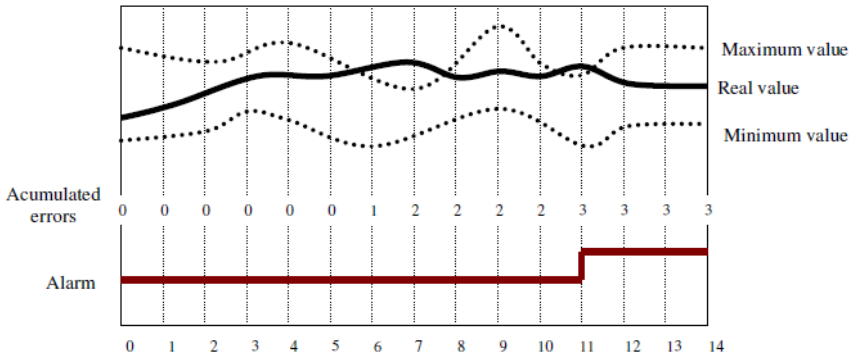


Fig. 2. Example of FIR fault detection using the enveloping method

The fault identification process is presented in Fig. 3.

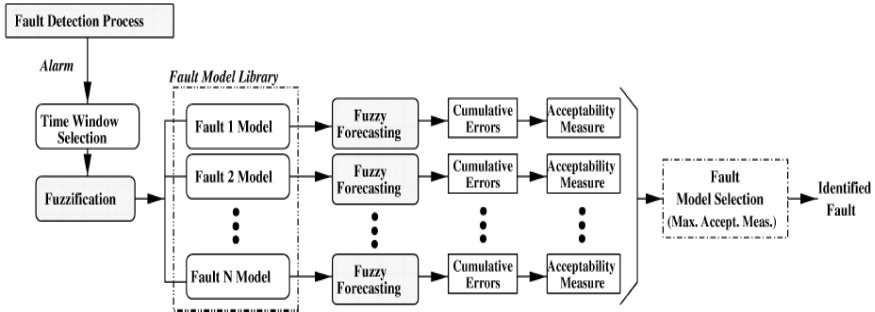


Fig. 3. FIR Fault Identification Process

In Fig. 3, the grey boxes represent FIR processes whereas the white boxes constitute the fault identification procedure. Once the alarm has been triggered because an abnormal behavior has been detected, a time window is selected. The size of the time window defines the number of prediction values that will be used in order to identify the fault that has been produced. Therefore, the time window guides the prediction during the identification process. A small size of the time window is desired because it implies fast model identification. For each fault model stored in the fault model library, a prediction of the size of the time window takes place using the FIR fuzzy forecasting process. The prediction errors produced during each of the forecasting processes are accumulated. Therefore, each fault model stored in the library has associated a cumulative error,  $Ia_i$ . This error is used to compute the model acceptability measure. The acceptability measure is a relative index ranking the models in terms of their ability to predict the new behavior of the system. This measure allows us, in a reliable way, to identify the fault that has occurred. It also offers guidance when the identification process faces additional problems, e.g. when the produced fault is not a foreseen fault and therefore is not available in the fault model library, or when two different models can be identified that both are able of explaining an observed fault. The acceptability measure of the  $i^{\text{th}}$  model,  $Q_i$ , is described by the following formula:

$$Q_i = C_i \cdot C_{rel,i} \tag{1}$$

where  $C_i$  is the partial acceptability measure of the  $i^{\text{th}}$  model and  $C_{rel,i}$  is a relative confidence that takes the dispersion between the  $C_i$  values into account.  $C_i$  is computed by use of the sum of cumulative errors for that particular model and the maximum number of local cumulative errors possible (depends on the size of the time window):

$$C_i = 1.0 - I_{a_i} / I_{a_{\max}} \quad (2)$$

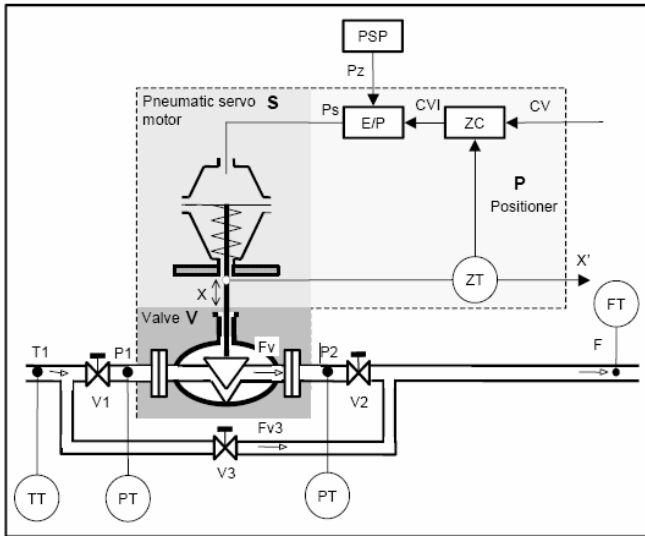
and the relative confidence is obtained by the equation:

$$C_{rel_i} = C_i / \sum_{k=1}^N C_k \quad (3)$$

The model with the largest acceptability measure is selected as the one that best represents the new behavior of the system, and therefore, the detected fault has been identified.

### 3 Case study: Fault Detection and Identification of a Valve Actuator

This section presents the results of applying VisualBlock-FIR to the DAMADICS actuator benchmark. The industrial actuator consists of a flow servo-valve driven by a smart positioner, as shown in Fig. 4.



**Fig. 4.** Schematic representation of the pneumatic servo-motor actuated control valve (Extracted from [8])

The actuator consists of the control valve (V), the pneumatic servomotor (S), and the positioner (P). These three main parts are composed by a set of basic measured physical values, such as flow sensor measurement (F), valve input pressure (P1), valve output pressure (P2), liquid temperature (T1), rod displacement (X) and external (flow or level) controller output (CV).

In order to test the robustness of the VisualBlock-FIR approach, two different faults have been simulated following the rules defined in DABLib for benchmark

purposes. The first fault correspond to fault F10, a servo-motor's diaphragm perforation caused by fatigue of diaphragm material, and the second one correspond to fault F1, a valve clogging that is a blocking servomotor rod displacement caused by an external mechanical event. Fault F10 corresponds to the pneumatic servomotor part and fault F1 to the control valve part. For both experiments, abrupt small and medium fault scenarios according to the DAMADICS benchmark definition are studied. Therefore, we are in fact dealing with four faults in this research, i.e., F10s, F10m, F1s, and F1m.

The first step is to obtain the FIR qualitative models that constitute the *fault library*. In this case, five different models should be identified, before VisualBlock-FIR is ready to perform the fault identification. Therefore, a FIR model for the correct behavior of the plant, as well as models for each type of diaphragm perforation behavior (small and medium) as well as for each type of valve clogging behavior (small and medium), are needed. For each of these scenarios, the system is simulated across 700 seconds of simulation time with a sampling rate of 0.25 seconds, generating 2800 data points. The first 100 seconds (400 data points) are used by FIR to identify the model, and the final 600 seconds (2400 data points) are used to verify the model. All the models are SISO, with CV as input variable and X as output variable. The mean square error, *MSE*, in percentage is used to compute the prediction error, as described in equation 4, where  $y(t)$  is the real signal and  $\hat{y}(t)$  is the predicted signal.

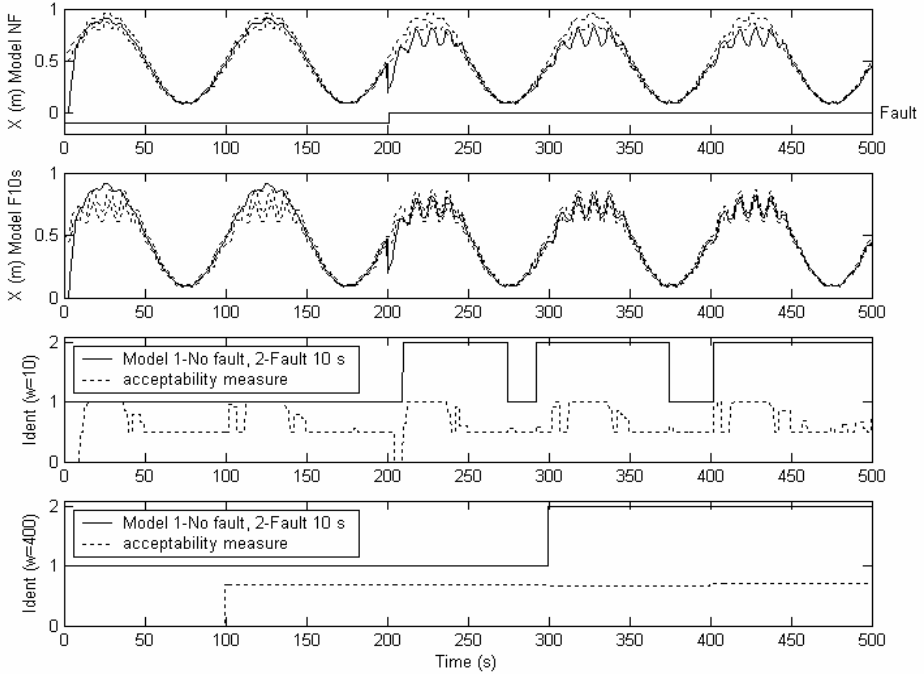
$$MSE = \frac{E\left[(y(t) - \hat{y}(t))^2\right]}{y_{\text{var}}}.100\% \quad (4)$$

An MSE of  $4.99e^{-2}\%$  is obtained for the non-fault model when it is used to predict the test data set. Errors of  $3.22e^{-2}\%$  and  $2.91e^{-2}\%$  are obtained when using the F1s and F1m models, respectively, to predict the test data set. Finally, MSEs of  $2.11e^{-1}\%$  and  $2.34e^{-1}\%$  are obtained when using the F10s and F10m models, respectively, to predict the test data set. These prediction errors are very low, meaning that the models obtained represent accurately each of the system's behaviors. The FIR models (masks and behavior matrices) of the four faults as well as the model of the valve working properly are stored in the fault model library to be used during the identification phase of VisualBlock-FIR.

### 3.1 Diaphragm Perforation Fault (F10)

Small and medium diaphragm perforations are introduced at time 200 seconds. Fig. 5 and 6 show the results of the detection process of VisualBlock-FIR for small (F10s) and medium (F10m) fault sizes, respectively. As can be seen from the top plot of both figures, the detection is performed almost instantaneously when the fault occurs. In this case, it is defined that three cumulative errors are needed in order to trigger an alarm. Therefore, for faults F10s and F10m, only 0.75 seconds are needed for VisualBlock-FIR to determine that the valve is not working properly. At time 200.75 seconds an alarm is activated as shown in the top plot of Fig. 5 and 6 (labeled with the word "Fault"). Notice that during the first 200 seconds the predicted signal of the model without fault is completely inside the enveloping interval (top plot of both figures) whereas the predicted signals of F10s and F10m faults are outside that

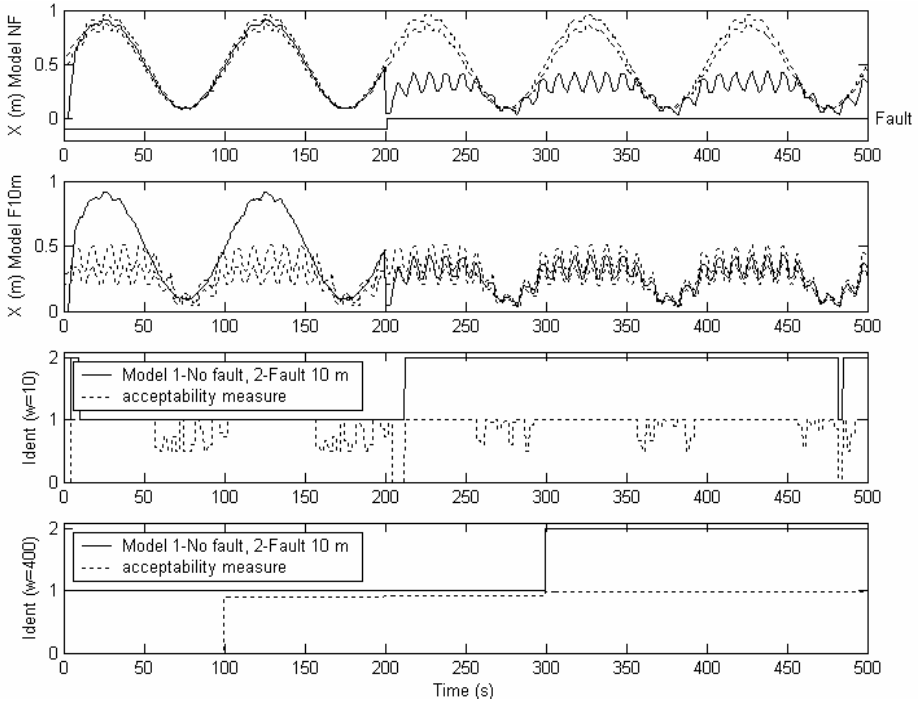
interval (second plot of both figures). After second 200 this behavior changes completely, now the predicted signals that are inside the enveloping interval are the ones obtained by means of the fault models. Notice however that the lower curves of the signals are exactly the same for both (fault and non-fault) system behaviors. Therefore, these segments of the signal are always inside the envelope.



**Fig. 5.** Fault F10s detection by means of VisualBlock-FIR

Once the alarm has been triggered the identification process starts (see Fig. 3). In this case a time window of 10 data points (corresponding to 2.5 seconds) is used to determine the fault that has occurred. The third plot of both figures show the identification process when a time window of 2.5 seconds is used. After fault detection at time 200.75 seconds, the identification process concludes that the model representing the F10s fault is the one that best corresponds with the actual behavior of the system. Notice however that the identification process fails twice during time periods of 18 seconds and 25 seconds when it decides that the model without fault is the one that represents best the observed behavior of the system. Looking closer at the two periods when the model identification fails, it can be seen that they correspond to the two valleys of the signal, i.e., time segments that both models can predict well because the fault and non-fault behaviors cannot be distinguished during these periods. The lower plot of Fig. 5 shows the identification process when a time window of 400 data points (corresponding to 100 seconds, i.e. a complete signal cycle), is being used. As can be seen, the identification process does an excellent work; during the last 300 seconds it

identifies the F10s model as the one that best follows the system behavior. Notice that the identification of the fault is produced at time 300 seconds only because the time window is now 100 seconds wide. Notice that during the first 99 seconds the acceptability measure assumes the default value of 0, until the acceptability measure is computed for the first time after 100 seconds (once the time window period has passed).



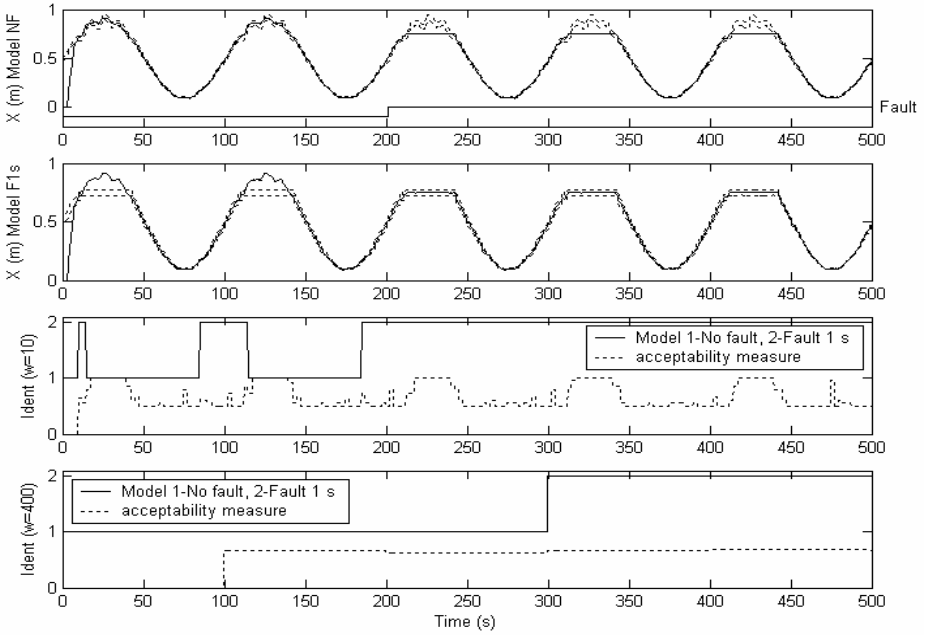
**Fig. 6.** Fault F10m detection by means of VisualBlock-FIR

The same analysis can be performed for the diaphragm perforation medium fault, F10m, shown in Fig. 6. In this case, the identification process with a time window of 10 data points (2.5 seconds) is almost perfect. Only one small mistake of a few seconds duration is encountered. When the time window is increased in such a way that it covers a complete signal cycle, the identification is done properly. Notice that in this case, the signals measured faultless operation of the system and during the presence of a F10m fault are quite different, making the identification task of the VisualBlock-FIR FDS easier.

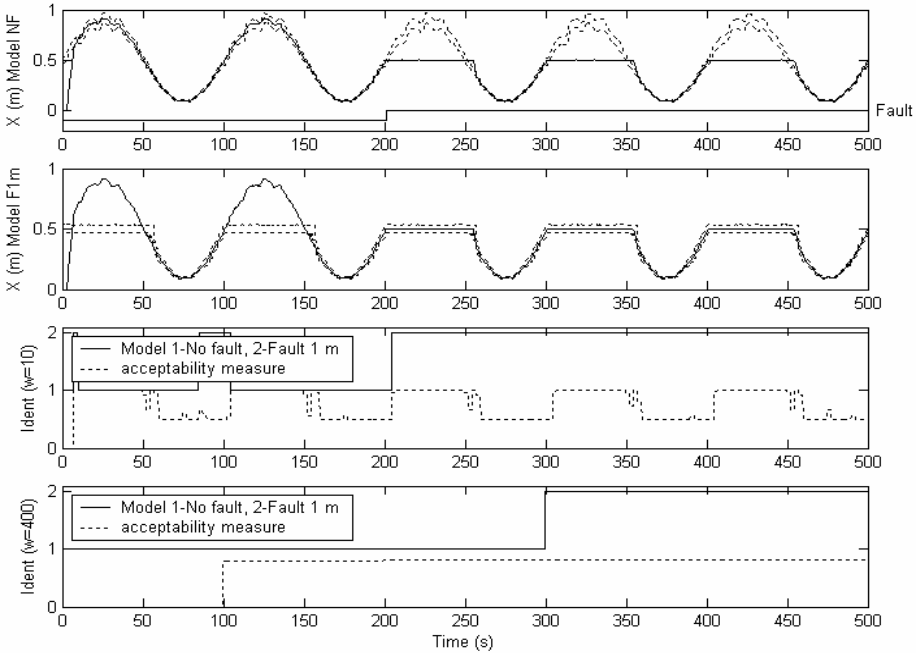
### 3.2 Valve Clogging Fault (F1)

Small and medium valve clogging faults are introduced at time 200 seconds. Fig. 7 and 8 show the results of the detection process of VisualBlock-FIR for small (F1s) and medium (F1m) fault sizes, respectively. In both cases, the valleys of the signals are essentially the same for both system behaviors, i.e. with and without a fault,





**Fig. 7.** Fault F1s detection by means of VisualBlock-FIR



**Fig. 8.** Fault F1m detection by means of VisualBlock-FIR

making the identification phase trickier. The two top plots of Fig. 7 and 8 show that the predictions obtained using the model of the system working properly as well as the models of the F1s and F1m faults are very accurate. As has happened in the case of the F10 faults, the detection is performed almost instantaneously when the fault occurs, as shown in the top plots of both figures.

The third plot of Fig. 7 and 8 shows that the fault identification phase of VisualBlock-FIR has three and two errors, respectively, during the first 200 seconds, where the system has no fault. However, the identification is performed very well in the period where the system presents the F1s and F1m faults. In this case, a time window of 10 data points (2.5 seconds) is used. When the time window is increased to cover a complete signal cycle, i.e., 400 data points (100 seconds), the identification is done properly as shown in the lower plot of Fig. 7 and 8.

Notice that in this research we have decided to activate the identification process of VisualBlock-FIR from the beginning, in order to better study its performance. However when VisualBlock-FIR is used in a real-world situation, it is recommended to start the identification process only after the detection phase has triggered an alarm that a fault has occurred, i.e. in the experiment at hand at second 200.75. Such a procedure would eliminate the false alarms that occur in the early phase.

From the experiments shown in Fig. 5, 6, 7, and 8, it can be concluded that VisualBlock-FIR is performing a good job detecting in less than 1 second that a fault has occurred. It also obtains good results in the identification phase, due to the fact that only 2.5 seconds are needed to identify the problem after the alarm went off. In a real-world situation, the identified fault would lead to an intervention in order to either correct the fault or, if this cannot be done, to shut down the system.

## 4 Conclusions

The VisualBlock-FIR fault diagnosis system is presented for the first time as a tool for detection and identification of faults for non-linear plants. The detection and identification processes are introduced in this article and are applied to tackle faults of the DAMADICS benchmark problem. Four different faults in the pneumatic servomotor part and the control valve part are detected and identified in an accurate way.

## Acknowledgments

This research was supported by the Consejo Interministerial de Ciencia y Tecnología under project TIN2006-08114.

## References

1. Basseville, M., Nikiforov, I.: *Detection of Abrupt Changes: Theory and Applications*. Prentice Hall Information and Systems Science Series, USA (1993)
2. Puig, V., Stancu, A., Escobet, T., Nejjari, F., Quevedo, J., Patton, R.J.: Passive robust fault detection using interval observers: Application to the DAMADICS benchmark problem. *Control Engineering Practice* 14, 621–633 (2006)

3. Previdi, F., Parisini, T.: Model-free actuator fault detection using a spectral estimation approach: the case of the DAMADICS benchmark problem. *Control Engineering Practice* 14(6), 635–644 (2006)
4. Crespo, A.: Real-Time Expert Systems. In: Boullart, L., Krijgsman, A., Vingerhoeds, R.A. (eds.) *Applications of Artificial Intelligence in Process Control*, Pergamon Press, UK (1993)
5. Kandel, A. (ed.): *Fuzzy Expert Systems*. CRC Press, USA (1992)
6. Witczak, M., Korbicz, J., Mrugalski, M., Patton, R.J.: A GMDH neural network-based approach to robust fault diagnosis: application to the DAMADICS benchmark problem. *Control Engineering Practice* 14(6), 671–683 (2006)
7. Witczak, M., Patton, R.J., Korbicz, J.: Fault detection with observers and genetic programming: application to the DAMADICS benchmark problem. In: *5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes - SAFEPROCESS 2003*, Washington, USA, 2003, pp. 1203–1208 (2003)
8. Calado, J.M.F., Carreira, F.P.N.F., Mendes, M.J.G.C., Sá da Costa, J.M.G., Bartys, M.: Fault detection approach based on fuzzy qualitative reasoning applied to the DAMADICS benchmark problem. In: *5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes - SAFEPROCESS 2003*, Washington, USA, pp. 1179–1184 (2003)
9. Rzepiejewski, P., Syfert, M., Jegorov, S.: On-line actuator diagnosis based on neural models and fuzzy reasoning: The DAMADICS benchmark study. In: *5th IFAC Symposium on Fault Detection Supervision and Safety of Technical Processes - SAFEPROCESS 2003*, Washington, USA, pp. 1083–1088 (2003)
10. Uppal, F.J., Patton, R.J., Witczak, M.: A hybrid neuro-fuzzy and de-coupling approach applied to the DAMADICS benchmark problem. In: *5th IFAC Symposium on Fault Detection Supervision and Safety of Technical Processes - SAFEPROCESS 2003*, Washington, USA, pp. 1059–1064 (2003)
11. Law, A., Kelton, W.: *Simulation modeling and analysis*, 2nd edn. McGraw-Hill, New York (1991)
12. Nebot, A.: *Qualitative modeling and simulation of biomedical systems using FIR*, Ph.D., Univ. Polit. Catalunya (1994)

# Sliding Mode Control of a Hydrocarbon Degradation in Biopile System Using Recurrent Neural Network Model

Ieroham Baruch<sup>1</sup>, Carlos-Roman Mariaca-Gaspar<sup>1</sup>, Israel Cruz-Vega<sup>1</sup>,  
and Josefina Barrera-Cortes<sup>2</sup>

CINVESTAV-IPN, Ave. IPN No 2508,  
A.P. 14-740, Mexico D.F., C.P. 07360, MEXICO

<sup>1</sup> Department of Automatic Control  
{baruch,cmariaca,icruz}@ctrl.cinvestav.mx

<sup>2</sup> Department of Biotechnology and Bioengineering  
jbarrera@mail.cinvestav.mx

**Abstract.** This paper proposes the use of a Recurrent Neural Network (RNN) for modeling a hydrocarbon degradation process carried out in a biopile system. The proposed RNN model represents a Kalman-like filter and it has seven inputs, five outputs and twelve neurons in the hidden layer, with global and local feedbacks. The learning algorithm is a modified version of the dynamic Back-propagation one. The obtained RNN model is simplified and used to design a Sliding Mode Control (SMC). The graphical simulation results of biopile system approximation, obtained via RNN model learning and the designed process SMC exhibited a good convergence, and precise system reference tracking.

## 1 Introduction

The Recent advances in understanding of the working principles of artificial Neural Networks (NN) has given a tremendous boost to identification and control tools of nonlinear systems, [1]. The main network property namely the ability to approximate complex non-linear relationships without prior knowledge of the model structure makes them a very attractive alternative to the classical modeling and control techniques. This property has been proved by the universal approximation theorem [2]. Among several possible network architectures the ones most widely used are the Feedforward NN (FFNN) and the Recurrent NN (RNN). In a FFNN the signals are transmitted only in one direction, starting from the input layer, subsequently through the hidden layers to the output layer, which requires applying a tap delayed global feedbacks and a tap delayed inputs to achieve a nonlinear autoregressive moving average neural dynamic plant model. A RNN has local feedback connections to some of the previous layers. Such a structure is suitable alternative to the FFNN when the task is to model dynamic systems, and the universal approximation theorem has been proved for RNN too. The preference given to RNN identification with respect to the classical methods of process identification is clearly demonstrated in the solution of

the “bias-variance dilemma” [2]. In [3] a comparative study of linear, nonlinear and neural-network-based adaptive controllers for a class of fed-batch baker’s and brewer’s yeast fermentation is done. The paper proposed to use the method of neural identification control, given in [1], and applied FFNNs (multilayer perceptron and radial basis functions NN). The proposed control gives a good approximation of the nonlinear plants dynamics, better with respect to the other methods of control, but the applied static NNs have a great complexity, and the plant order (and plant structure, especially for MIMO plants) has to be known. The application of RNNs could avoid these problems and could reduce significantly the size of the applied NNs. Furthermore, in biotechnology there exists a great variety of processes with incomplete information where analytical models description is missing. One of them is the hydrocarbon degradation process of contaminated with petroleum soils in biopile system, [4]. The control and manipulation of the hydrocarbon removal by a bio-stimulation process is a complex task in itself due to the great variety of native micro-organisms involved. Considering the lack of knowledge about each micro-organism’s metabolism and their interactions, considering the limited information that is provided by each single degradation process, it is of our concern to develop a model that might correlate the behavior of the response variables such as pH, humidity, carbon dioxide production as well Total Petroleum Hydrocarbons (TPHs) concentration and others, all of these, along the time at which the biodegradation process occurs.

In some early papers, [5], [6], [7], the state-space approach is applied to design a RNN, defining a Jordan canonical two or three layer Recurrent Trainable Neural Network (RTNN) model and a Backpropagation (BP) algorithm of its learning. This RNN model is a parametric one and it serves as state and parameter estimator, which permits to use the estimated states and parameters directly for process control. In [6] this general RTNN approach is applied in an indirect and direct neural control schemes for identification and control of continuous wastewater treatment fermentation bioprocess, where unfortunately the plant and measurement noises affected the control precision.

In the proposed paper we go ahead extending the topology of this RNN with local and global feedbacks. This topology has been applied to predict output variables of various bioprocesses like fed-batch fermentation of Bt [8], osmotic dehydration process, [9] and finally hydrocarbon degradation profiles in biopile, [10]. This topology with built in output filter is capable to decrease measurement noise and to correlate different process measurements in order to obtain a complete process neural model learned by the BP algorithm. Then this RNN model is simplified and could be used for different control system design methods, like the Sliding Mode Control (SMC), so to achieve the control objectives depending on the available process measurements.

## 2 Recurrent Neural Model Description

Block-diagrams of the extended RNN topology and its adjoint, are given on Fig. 1, and Fig. 2. Following Fig. 1, and Fig. 2, we could derive the Backpropagation algorithm of its learning based on the RNN topology using the diagrammatic method of [11]. The RNN topology and learning is described in vector-matrix form as:

$$\begin{aligned}
 X(k+1) &= A_1 X(k) + BU(k) - DY(k); & (1) \\
 Z(k) &= \Gamma [X(k)]; Z_1(k) = C Z(k) & (2) \\
 V(k+1) &= Z_1(k) + A_2 V(k); & (3) \\
 Y(k) &= \Phi[V(k)] & (4) \\
 A_1 &= \text{block-diag} (A_{1,i}), |A_{1,i}| < 1; & (5) \\
 A_2 &= \text{block-diag} (A_{2,i}), |A_{2,i}| < 1 & (6) \\
 W(k+1) &= W(k) + \eta \Delta W(k) + \alpha \Delta W_{ij}(k-1) & (7) \\
 E(k) &= T(k) - Y(k) & (8) \\
 R_1 &= E(k) [1 - Y^2(k)] & (9) \\
 \Delta C(k) &= R_1 Z^T(k) & (10) \\
 \Delta A_2(k) &= R_1 V^T(k) & (11) \\
 R &= C^T(k) E(k) [1 - X^2(k)] & (12) \\
 \Delta B(k) &= R U^T(k) & (13) \\
 \Delta D(k) &= R Y^T(k) & (14) \\
 \Delta A_1(k) &= R X^T(k-1) & (15) \\
 \text{Vec}(\Delta A_2(k)) &= R_1 \circ V(k) & (16) \\
 \text{Vec}(\Delta A_1(k)) &= R \circ X(k-1) & (17)
 \end{aligned}$$

Where:  $Y, X, U$  are output, state and input vectors with dimensions  $l, n, (l+m)$ , respectively; here  $U^T = [U_1; T]$ , where  $U_1$  is the real plant input vector with dimension  $m$ ;  $T$  is the plant output vector with dimension  $l$ , considered as a RNN reference;  $A_1, A_2$  are  $(n \times n)$  and  $(l \times l)$ - local feedback block-diagonal weight matrices respectively, defined by (5), (6);  $B = [B_1; B_2]$  and  $C$  are  $[n \times (l+m)]$  and  $(l \times n)$ - weight matrices, where  $B_1$  corresponds to  $U_1$  and  $B_2$  corresponds to  $T$ ;  $D$  is a  $(n \times l)$  global output closed loop matrix;

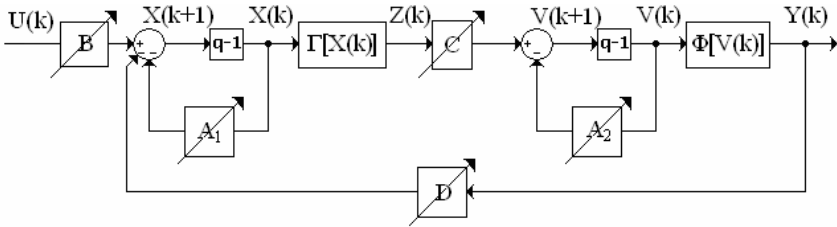


Fig. 1. Block diagram of the RNN model

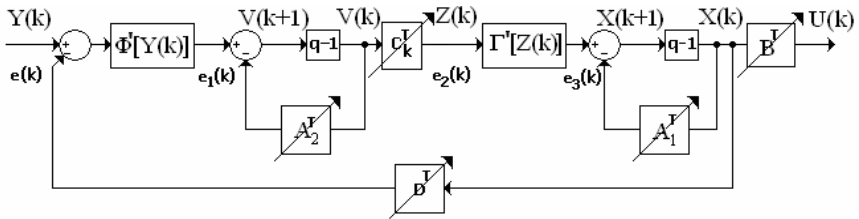


Fig. 2. Block diagram of the adjoint RNN model

$\Gamma[\cdot], \Phi[\cdot]$  are vector-valued tanh-activation functions;  $W$  is a general weight, denoting each weight matrix ( $C, D, A_1, A_2, B$ ) in the RNNM model, to be updated;  $\Delta W$  ( $\Delta C, \Delta D, \Delta A_1, \Delta A_2, \Delta B$ ), is the weight correction of  $W$ ;  $\eta, \alpha$  are learning rate parameters;  $\Delta C, \Delta A_2$  are weight corrections of the  $(1 \times n)$  learned matrix  $C$  and the  $(1 \times 1)$  learned diagonal matrix  $A_2$ ;  $R_1$  and  $R$  are auxiliary vector variables;  $\Delta B, \Delta D$  are weight corrections of the  $[n \times (l+m)]$  learned matrix  $B$  and the  $(n \times l)$  learned matrix  $D$ ; the diagonals of the matrices  $A_1, A_2$  are denoted by  $\text{Vec}(\cdot)$  and equations (16), (17) represents its learning as an element-by-element vector products. The stability of the RNN Model (RNNM) is assured by the activation functions bounds and by the local stability bound conditions, given in (5), (6). As it could be seen in Fig. 1, the first part of the RNNM, given by the statements (1), (2) (without the global feedback entry) represents the plant model, and the second part, given by the statements (3), (4) represents an output filter part. The complete RNN structure is in fact a full order state observer (Kalman – like filter) where the balance between the reference and feedback parts ( $B_2T \rightarrow DY$ ) is achieved during the learning, when ( $E \rightarrow 0$ ). A simplified version of this RNN model, containing only the plant part of the model is used as a base for the design of a SMC. Further, it will be show that the indirect adaptive neural control, used in [6], could be derived as a SMC defining the Sliding Surface (SS) with respect to the plant output.

### 3 A Sliding Mode Control Systems Design

The block diagram of the control scheme is shown on Fig. 3. It contains identification and state estimation RNNM and a SMC. The stable nonlinear plant is identified by a RNNM with topology, given by equations (1)-(6) learned by the stable BP-learning algorithm, given by equations (7)-(17), where the identification error (8) tends to zero. The simplification and linearization of the identified RNNM (1), (2), omitting the  $DY(\cdot)$  and  $B_2T(\cdot)$  parts, leads to the following local linear plant model:

$$X(k+1) = A_1X(k) + BU(k) \tag{18}$$

$$Z(k) = F X(k); F = C \Gamma'(Z) \tag{19}$$

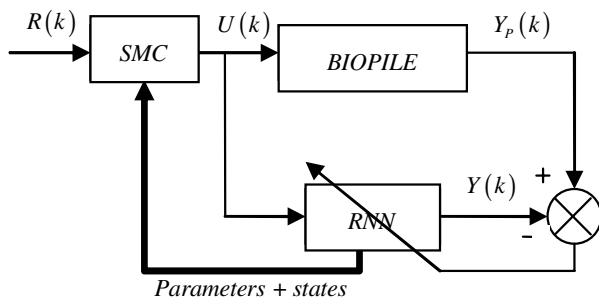


Fig. 3. Block diagram of the closed-loop system

Where  $\Gamma'(\cdot)$  is the derivative of the activation function and  $l = m$ , is supposed. Let us define the following SS as an output tracking error function:

$$S(k+1) = E(k+1) + \sum_{i=1}^p \gamma_i E(k-i+1); |\gamma_i| < 1 \tag{20}$$

Where:  $S(\cdot)$  is the Sliding Surface Error Function (SSEF);  $E(\cdot)$  is the systems output tracking error;  $\gamma_i$  are parameters of the desired SSEF;  $p$  is the order of the SSEF. The tracking error and its iterate are defined as:

$$E(k) = R(k) - Z(k); E(k+1) = R(k+1) - Z(k+1) \tag{21}$$

Where  $R(k)$ ,  $Z(k)$  are 1-dimensional reference and output vectors. The objective of the sliding mode control systems design is to find a control action which maintains the systems error on the sliding surface which assure that the output tracking error reaches zero in  $p$  steps, where  $p < n$ . So, the control objective is fulfilled if:

$$S(k+1) = 0 \tag{22}$$

Now, let us to iterate (19) and to substitute (18) in it so to obtain the input/output local plant model, which yields:

$$Z(k+1) = F X(k+1) = F [AX(k) + BU(k)] \tag{23}$$

From (20), (21) and (22) it is easy to obtain:

$$R(k+1) - Z(k+1) + \sum_{i=1}^p \gamma_i E(k-i+1) = 0 \tag{24}$$

The substitution of (23) in (24) gives:

$$R(k+1) - FAX(k) - FB U(k) + \sum_{i=1}^p \gamma_i E(k-i+1) = 0 \tag{25}$$

As the local approximation plant model (18), (19), is controllable, observable and stable, [5], the matrix  $A_1$  is diagonal, and  $l = m$ , then the matrix product  $(FB)$  is non-singular, and the plant states  $X(k)$  are smooth non-increasing functions. Now, from (25) it is easy to obtain the equivalent control capable to lead the system to the sliding surface which yields:

$$U_{eq}(k) = (FB)^{-1} [-FAX(k) + R(k+1) + \sum_{i=1}^p \gamma_i E(k-i+1)] \tag{26}$$

Following [12], the SMC avoiding chattering is taken using a saturation function instead of sign one. So the SMC takes the form:



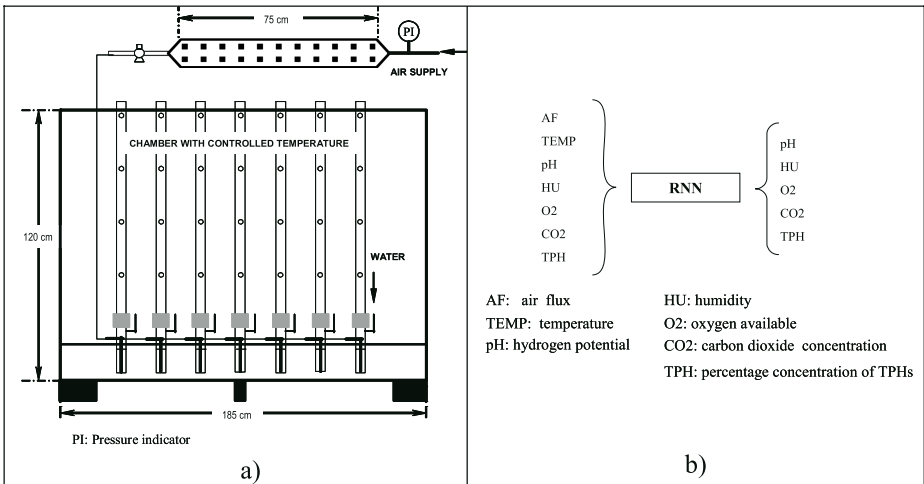
$$U^*(k) = \begin{cases} U_{eq}(k), & \text{if } \|U_{eq}(k)\| < U_0 \\ -U_0 \frac{U_{eq}(k)}{\|U_{eq}(k)\|}, & \text{if } \|U_{eq}(k)\| \geq U_0. \end{cases} \quad (27)$$

The SMC substituted the multi-input multi-output coupled high order dynamics of the linearized plant with desired decoupled low order one.

#### 4 Description of the Hydrocarbon Biodegradation Experiment

Bioremediation in biopile system is an *ex-situ* Solid Substrate Fermentation (SSF) technology, based on the ability of micro-organisms to degrade pollutant hydrocarbon compounds [4]. The often used bio-stimulation technique consists on the activation of the native soil micro-organisms by addition of nutrients, water, oxygen (for aerobic process) and a bulking agent that let it to improve the oxygen supplied to the microorganisms [13]. SSF takes place in the absence of free water, so it offers the advantage of reducing the place and cost requirements [14]. The SSF disadvantage consists of the complexity and heterogeneity of the solid matrix, which makes quite difficult the measurement and control of process variables. The interest of the biopile technology is an inherent temperature increase inside the biopile - from the centre to the surface, which favors the sequential development of a microbial population growth associated to the temperature profile and residual pollution [15]. Temperature increase can reach 60°C, so it is frequently controlled by an air flux supplied to the biopile columns. Besides, controlling the temperature, the air flux is a source of fresh oxygen to the microorganisms. The next environmental conditions are recommended for an adequate hydrocarbon biodegradation in biopile system: pH  $\approx$  7; humidity at 50-60% of the Water Holding Capacity (WHC) of soil; average temperature of 30°C. It is important to supply an adequate air flux, since a low one could not be enough for satisfying the microbial requirements, but a high air one could dry the solid matrix. In this study, it is used a crumb-limose soil from a site polluted near a refinery in México. The pollution of 165000 ppm, consist on different residues of crude oil process and refining. The soil was dried and blended with ocorn used as a bulking agent 10:1 (% v/v), which was milled and sterilized. The moisture was adjusted at 60% of water retention capacity, and C:N:P ratio at 100:10:1 according to analyses done. Tergitol 1% (p/p) was used as surfactant to enhance contaminant desorption from soil. The equipment used is shown on Fig. 4 a, and the Input/Output full RNN learning pattern in shown on Fig. 4 b. The biopile system consists of twenty one columns (1.0 m height x 3.81 cm i.d.), constructed to allow the monitoring through 28 days, almost each other day. Each column has sample ports located at the sites every 25 cm, and was fitted with water vessels to humidify the air entering the columns. The columns were housed in a chamber provided with temperature control, and the air was supplied at a constant pressure via a manifold.

The experiment consists of seven sets of fermentation data taken for different air flux (180, 360, 450 and 540 ml/min) and different temperature (20 and 40°C). The duration of the bioremediation process depends on the volume of the soil under treatment and the type and concentration of the contaminants in it. In our case 28 days are



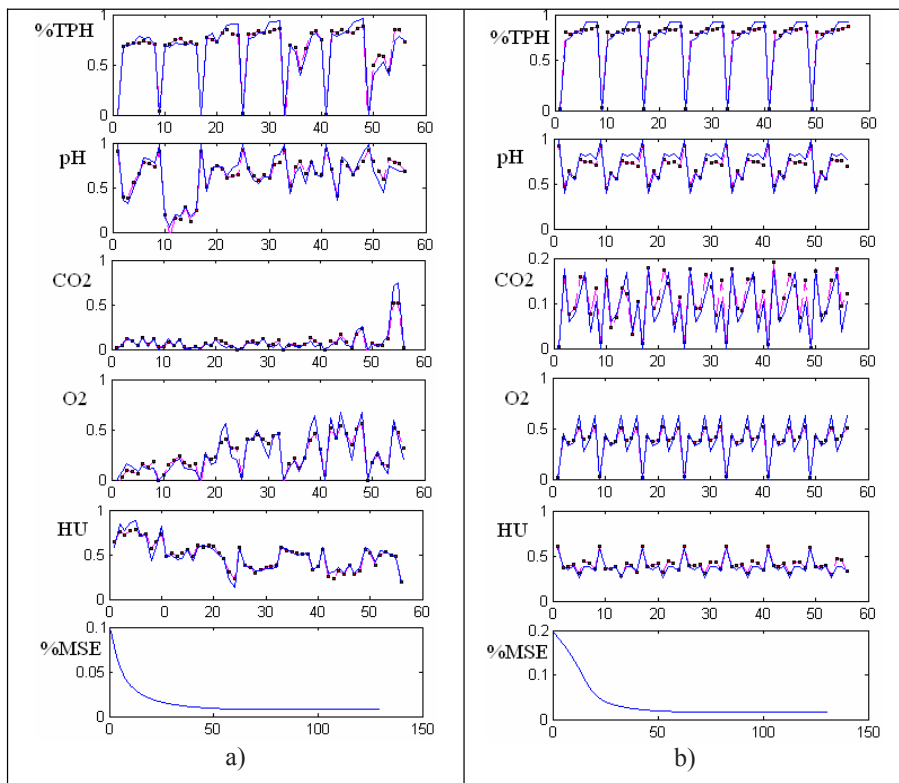
**Fig. 4. a)** Sketch of the biopile system; **b)** Learning pattern of the full RNN model

sufficient to degrade 60% of the contaminants which is considered sufficient for our experiment. The evolution of the hydrocarbon removal was evaluated from solid samples periodically extracted from the biopile for analysis of pH (potentiometer), humidity (gravimetric method), oxygen consumption and carbon dioxide production - by gas chromatography, TPHs - by infrared spectroscopy, following soxhlet extraction with dichloromethane (EPA Method 3540C).

## 5 Experimental Results of Bioprocess Identification

The graphical results of the experimental neural biodegradation process identification are given on Fig. 5 a – for RNN learning, and on Fig. 5 b – for RNN generalization. The Input Learning Pattern (ILP) proposed is conformed by the: ILP(AF, TEMP, pH, HU, O<sub>2</sub>, CO<sub>2</sub>, TPH- total petroleum hydrocarbons). The Output Learning Pattern (OLP) includes: OLP(pH, HU, O<sub>2</sub>, TPH). The RNN used for modeling and identification of the hydrocarbon degradation process in biopile system has seven inputs, twelve neurons in the hidden layer and five outputs. The number of neurons (twelve) in the hidden layer was determined in an experimental way, according to the Mean Square Error (MSE%) of learning. The learning algorithm is a version of the dynamic BP one, specially designed for this RNN topology. The described above learning algorithm is applied simultaneously to 7 degradation kinetic data sets (patterns), realized below different conditions of air flow and temperature in the ranges 180-540 mi/min and 25-50°C, and containing 8 points each one.

The experimental data were normalized in the range 0-1 due to the great difference in magnitude between them. The 7 data sets are considered as an epoch of learning, containing 56 points. After each epoch of learning, the 7 pattern sets are interchanged



**Fig. 5.** Graphical results of experimental biodegradation process identification; **a)** graphical results of RNN learning (%TPH, pH, CO<sub>2</sub>, O<sub>2</sub>, HU, and MSE%); **b)** graphical results of RNN generalization (%TPH, pH, CO<sub>2</sub>, O<sub>2</sub>, HU, and MSE%)

in an arbitrary manner from one epoch to another. An unknown kinetic data set, repeated 7 times, is used as a generalization data set. The learning is stopped when the MSE% of learning reached values below 2%, the MSE% of generalization reached values below 2%, and the relationship  $|\Delta W_{ij}(k)|/|W_{ij}(k)| \cdot 100\%$  reached values below or equal of 2% for all updated parameters. This error was attained after 131 epochs of learning. The graphical results shown on Fig. 5 a. compared the experimental data for the 7 degradation kinetics with the outputs of the RNN during the last epoch of learning. The variables compared and plotted subsequently for the last epoch of learning are % degradation in TPH, pH, carbon dioxide (CO<sub>2</sub>), oxygen available (O<sub>2</sub>), % of humidity (HU) and the mean square error (MSE%) given for 131 epochs of learning. The learning rate is 0.9, the momentum rate is 0.8, the epoch size contains 56 points, the convergence is obtained after 131 epochs of learning. The variables shown are: %TPH, pH, CO<sub>2</sub>, O<sub>2</sub>, HU, MSE% of learning. The final MSE% of learning is below 2%. The generalization of the RNN was carried out reproducing a degradation kinetics which is not included in the training process. This degradation process was carried

out at AF = 360 ml/min and temperature of 20°C. The operational conditions of this degradation process are in the range of operational conditions studied. The generalization results shown on Fig. 5 b. compare the experimental data for the one unknown degradation kinetics (repeated 7 times so to maintain the epoch size) with the output of the RNN. The experimental data (continuous line) and the RNN outputs (data point line) and are plotted subsequently for the last epoch of generalization. The variables shown are: %TPH, pH, CO<sub>2</sub>, O<sub>2</sub>, HU, MSE% of generalization. The final MSE% of RNNM generalization is below 2%.

### 6 Simulation Results of Bioprocess SMC

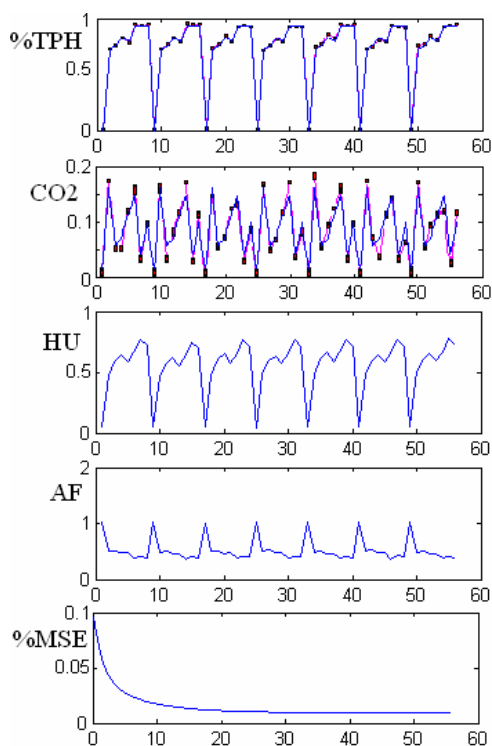
A simplified process model, extracted from the complete RNNM, has been used to design a SMC system. The RNN particular model has 2 inputs (AF, HU), two outputs (%TPH, CO<sub>2</sub>) and 12 states. In that reduced model, depending on the available measurements, the input and output patterns are chosen as: ILP(AF, HU, CO<sub>2</sub>, TPH); OLP(CO<sub>2</sub>, TPH). The graphical simulation results of the controlled system outputs (%TPH, CO<sub>2</sub>), the control variables (AF, HU), and the MSE% are given on Fig. 6. The obtained MSE% of control in the end of the process is below 1% and it is given in Table 1 for 20 runs of the control program. The two system set points (continuous line) are compared with the two plant outputs (%TPH, CO<sub>2</sub>) (pointed line) and are plotted subsequently for seven sets of set point data. The control variables shown are: AF, HU. However the lost of water is pretended to be compensated by the wet saturated air flux with controlled humidity introduced, which could accelerates the bioremediation process in the biopile system. The behaviour of the control system in the presence of 10% white gaussian noise added to the plant outputs could be studied acumulating some statistics of the final MSE% ( $\xi_{av}$ ) for multiple run of the control program (see Table 1). The mean average cost for all runs ( $\epsilon$ ) of control, the standard deviation ( $\sigma$ ) with respect to the mean value and the deviation ( $\Delta$ ) are given by the following formulas:

$$\epsilon = \frac{1}{n} \sum_{k=1}^n \xi_{av_k} ; \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n \Delta_i^2} ; \Delta = \xi_{av} - \epsilon \tag{28}$$

Where  $k$  is the run number and  $n$  is equal to 20. The mean and standard deviation values of process control, obtained, are respectively:  $\epsilon = 1.1682\%$ ;  $\sigma = 0.1276\%$ .

**Table 1.** Final MSE (%) of control ( $\xi_{av}$ ) for 20 runs of the control program

No	1	2	3	4	5
MSEC	1.106	1.0035	1.001	1.0951	0.93454
No	6	7	8	9	10
MSEC	1.146	1.3214	1.225	1.4721	1.1206
No	11	12	13	14	15
MSEC	1.3185	1.1544	1.1821	1.0316	1.1267
No	16	17	18	19	20
MSEC	1.1295	1.3268	1.1842	1.2858	1.1993



**Fig. 6.** Graphical results of the biodegradation process SMC (%TPH, CO<sub>2</sub>, AF, HU, MSE%)

## 7 Conclusions

The paper proposes a new Kalman like Filter (KF) closed loop topology of recurrent neural network for modeling a hydrocarbon degradation process carried out in a biopile system. The proposed KF RNN model has seven inputs, five outputs and twelve neurons in the hidden layer, with global and local feedbacks. The learning algorithm is a modified version of the dynamic Backpropagation one. The obtained RNN model issued parameters and states information appropriate for control systems design purposes. The obtained RNN model is simplified and used to design a sliding mode control. The simulation results obtained with the RNN model learning and control exhibit a good convergence and precise reference tracking. The MSE% of the RNN learning and generalization is below 2% and the MSE% of control is below 1%.

## Acknowledgements

This work was supported by CONACYT, Mexico through the project SEMARNAT-2002-C01-0154. The M.S. student Israel Cruz Vega and the Ph.D. student Carlos-Roman Mariaca-Gaspar are thankful to CONACYT for the scholarship received during their studies at the Department of Automatic Control, CINVESTAV-IPN, Mexico.

## References

1. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamic Systems Using Neural Networks. *IEEE Trans. Neural Networks* 1(1), 4–27 (1990)
2. Haykin, S.: *Neural Networks, a Comprehensive Foundation*, 2 edn., Section 2.13, pp. 84–89, Section 4.13, pp. 208–213. Prentice-Hall, Upper Saddle River (1999)
3. Boskovic, J.D., Narendra, K.S.: Comparison of Linear, Nonlinear and Neural-Network-Based Adaptive Controllers for a Class of Fed-Batch Fermentation Processes. *Automatica* 31, 817–840 (1995)
4. Alexander, M.: *Biodegradation and Bioremediation*. Academic Press, New York (1999)
5. Baruch, I.S., Flores, J.M., Nava, F., Ramirez, I.R., Nenkova, B.: An Advanced Neural Network Topology and Learning, Applied for Identification and Control of a D.C. Motor. In: *Proc. 1st Int. IEEE Symp. Intelligent Systems, IS 2002, Varna, Bulgaria*, vol. 1, pp. 289–295 (2002)
6. Baruch, I.S., Georgieva, P., Barrera-Cortes, J., Feyo de Azevedo, S.: Adaptive Recurrent Neural Network Control of Biological Wastewater Treatment. *International Journal of Intelligent Systems (Wiley Editors)* 20(2), 173–194 (2004)
7. Baruch, I., Barrera-Cortes, J., Hernandez, L.A.: A Fed-Batch Fermentation Process Identification and Direct Adaptive Neural Control with Integral Term. In: *Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972*, pp. 764–773. Springer, Heidelberg (2004)
8. Valdez-Castro, L., Baruch, I.S., Barrera-Cortes, J.: Neural Networks Applied to the Prediction of Fed-Batch Fermentation Kinetics of *Bacillus Thuringiensis*. *Bioprocess and Biosystems Engineering* 25, 229–233 (2003)
9. Baruch, I.S., Genina-Soto, P., Barrera-Cortes, J.: Predictive Neural Model of an Osmotic Dehydration Process. *Journal of Intelligent Systems* 14(2-3), 143–155 (2005) (Special Issue on Hybrid Intelligent Systems for Time Series Prediction)
10. De la Torre-Sanchez, R., Baruch, I.S., Barrera-Cortes, J.: Neural Prediction of Hydrocarbon Degradation Profiles Developed in a Biopile. *Expert Systems with Applications (Elsevier Editors)* 31, 283–389 (2006)
11. Wan, E., Beaufays, F.: Diagrammatic Method for Deriving and Relating Temporal Neural Networks Algorithms. *Neural Computations* 8, 182–201 (1996)
12. Young, K.D., Utkin, V.I., Ozguner, U.: A Control Engineer's Guide to Sliding Mode Control. *IEEE Trans. on Control Systems Technology* 7(3), 328–342 (1999)
13. Larsen, K., McCartney, D.: Effect of C.N. Ratio on Microbial Activity and N. Retention: Bench-Scale Study Using Pulp and Paper Biosolids. *Compost Science and Utilization* 8(2), 147–159 (2000)
14. Pandey, A.: Recent Process Developments in Solid-State Fermentation. *Process Biochemistry* 27, 109–117 (1992)
15. Joshua, R., Macauley, B., Mitchell, H.: Characterization of Temperature and Oxygen Profiles in Windrow Processing Systems. *Compost Science and Utilization* 6(4), 15–28 (1998)

# Knowledge Acquisition in Intelligent Tutoring System: A Data Mining Approach

Simone Riccucci<sup>1</sup>, Antonella Carbonaro<sup>2</sup>, and Giorgio Casadei<sup>1</sup>

<sup>1</sup> University of Bologna, Computer Science Department, Via Mura Anteo Zamboni 7,  
40121, Bologna, Italy

{riccucci,casadei}@cs.unibo.it

<http://www.cs.unibo.it/~{riccucci,casadei}>

<sup>2</sup> University of Bologna, Computer Science Department,  
Via Sacchi 3, 47023, Cesena, Italy

carbonar@csr.unibo.it

<http://www.csr.unibo.it/~carbonar>

**Abstract.** In the last years Intelligent Tutoring Systems have been a very successful way for improving learning experience. Many issues must be addressed until this technology can be defined mature. One of the main problems within the Intelligent Tutoring Systems is the process of contents authoring: knowledge acquisition and manipulation process is a difficult task because it requires specialized skills on computer programming and knowledge engineering. In this paper we propose a mechanism based on first order data mining to partially automate the process of knowledge acquisition. The knowledge has to be used in the ITS during the tutoring process for personalized instruction. Such a mechanism can be applied in Constraint Based Tutor and in the Pseudo-Cognitive Tutor.

## 1 Introduction

Intelligent Tutoring Systems are very useful tools to support and enhance the learning process in many fields. This kind of systems includes the necessary information for a real simulation of teaching activity: nowadays, most of the systems used in learning support, are a slight improvements of automated textbook. Furthermore, they do not embody any particular instructional approach, theory, or philosophy, other than the instructional approach that exist in the textbook on which the system is based.

On the other hand, ITSs can adapt their behaviour on the base of the domain and student models, approaching the benefits of one-on-one instruction. Many ITSs have been proved highly effective. PAT Algebra Tutor [1] was developed for use in High-School setting and is based on the ACT theory [2]. The main purpose of the system is teaching to apply mathematics learned at school to real world problems. An experiment was conducted on 470 students using this system and the experimental classes outperformed students in comparison classes by 15% on standardized tests and 100% on tests targeting the ITS objectives

(real life problems). LISP Tutor [3], a system for tutoring on LISP programming language, enabled students to cover more exercises in the same amount of time compared to students that did not use the tutor, which subsequently gave them an advantage. SQL-Tutor [4] is a knowledge-based teaching system which supports students learning SQL. It is based on Ohlssons theory of learning from performance errors. In an evaluation studies SQL-Tutor has been proved to be effective in performance increasing to solve problems.

Despite these instruments have been demonstrated very useful in the learning process, they suffer from the disadvantage of being difficult to construct especially from people who do not have notions of knowledge engineering and/or of computer programming. In fact nearly all these systems need for the presence of an expert in ITSs design and construction assisted by the expert of the domain being implemented. However, many efforts have been made in order to facilitate the authoring process of the ITSs like WETAS-CAS [5,6,7] for the ITS based on constraint model and CTAT [8,9] for the ITS based on ACT Cognitive Theory. We briefly review these systems and then propose a framework and algorithm to partially automate the process of knowledge acquisition based on multi relational data mining techniques.

## 2 Cognitive Tutor and CTAT

Cognitive tutors are based on the ACT-R theory of mind. The central principle of this theory is that the processes of thought can be modelled using declarative and procedural knowledge. Declarative knowledge corresponds to things we are aware we know and can usually describe to others. Examples of declarative knowledge include “Luigi Einaudi was the first president of the Italian Republic” and “The halting problem is not computable”. Procedural knowledge is knowledge which we show in our behaviour but which we are not conscious of. For example, no one can describe the rules by which we speak a language but we can do it. In ACT-R declarative knowledge is represented in structures called chunks whereas procedural knowledge is represented in productions. Thus chunks and productions are the basic building blocks of an ACT-R model.

Tutoring is achieved using a method known as model tracing. As the student works at the problem, the system traces her progress along valid paths in the model. If she makes a recognisable off-path action she is given an error message indicating what she has done wrong, or perhaps an indication of what she should do. If the action is identified as being off-path but cannot be recognised, she may only be told that it is incorrect, but not why.

Knowledge tracing is used to monitor the knowledge students have acquired from problems. A Bayesian procedure is used to estimate the probability that a production rule has been learned after each attempt. This assessment information is used to individualize problem selection and optimally route students through the curriculum.

Cognitive Tutor Authoring Tools (CTAT) consist of a set of instruments designed to allow programmers with more or less experiences to quickly realize a



“pseudo cognitive tutor”. The tools have been realized after long and deepened studies on the human-machine interactions and the fundamental principles of cognitive science, so as to realize an instrument easy to use but extremely effective. Essentially CTAT system provides widgets that are java objects used in graphical interface for supplying interaction between the student and the system. After we have designed the interface, interaction with such interface is needed in order to record the procedures carried out in resolving a problem. To the end of every interaction the system produces a behavioral graph that determines the procedure to solve a specific problem. In the generated graph there are both right behaviors that lead to the right solution and wrong behaviors that lead in a wrong state. For every state it will then be necessary to insert the relevant suggestions and feedbacks that will be given to the student while she interacts with the system. The difference between the “pseudo cognitive tutor” produced with CTAT and the full Cognitive Tutor is that in the former there are not generic production rules as in the latter but they have a specific instance of a problem solving procedure (the behavioral graph) allowing mimicking the behavior of a similar cognitive tutor. This fact on one side allows to quickly develop simple tutor, from the other it places limitations on the knowledge representation in more complex domain and, as emphasized in [9], it does not scale well to a tutor of great dimensions.

### 3 Constraint-Based Tutor and WETAS-CAS

This kind of ITS is based on Ohlssons theory of learning from performance errors. Constraint Based Modeling (CBM) focuses on faulty knowledge, realizing that it is not sufficient to describe what the student knows correctly. The basic assumption is that diagnostic information is not hidden in the sequence of students actions, but in the problem state the student arrived at. This assumption is supported by the fact that there can be no correct solution of a problem that traverses a problem state, which violates fundamental ideas, or concepts of the domain.

The constraint-based model proposed by Ohlsson represents both domain and student knowledge in the form of constraints, where the constraints represent the basic principles underlying the domain. A constraint is characterised by a relevance clause, and a satisfaction clause. The relevance clause is a condition that must be true before the constraint is relevant to the current solution. Once the relevance clause has been met, the satisfaction clause must be true for the solution to be correct.

For example, we can consider a person learning to drive. On approaching an intersection, she must consider many factors regarding who gives way and decide whether or not to stop. Such pieces of knowledge relate, among other things, to the driving rules of the country she is in. If we are in Italy for example, one such rule is that “at uncontrolled intersections, traffic on the right has right-of-way”. Now, as our driver approaches an uncontrolled intersection, she must consider whether or not to give way. A constraint for the above situation might be:

```

IF uncontrolled intersection
AND car approaching from right
THEN give way

```

WETAS and CAS are systems developed for authoring a Constraint-based Tutor: the former provides all the domain-independent components for a text-based ITS, including the user interface, pedagogical module and student modeller; the latter is a system for acquiring the domain model in semi automatic way.

In the CAS system the constraints are acquired through a four stages process: the first phase consists in representing the base domain concepts and the relations between them by means of an ontology; in the second phase the syntactic constraints are extracted through a procedure described in [10]; in the third phase author provides a set of problem-solution to the system with  $n$  solutions for every problem and semantic constraints are extracted as described in [7]; in the last phase the constraint set is validated with the assistance of the domain expert, where the expert should label the system generated examples as correct or incorrect.

The domain constraint acquisition system, even if is currently limited to declarative knowledge, has been demonstrated very effective in two domains but it has not been evaluated for its generality in other types of domains. However, a new functionality is being developed that will allow to acquire also procedural knowledge.

## 4 Knowledge Acquisition: Preliminary Concepts

We use a framework of multirelational data mining described in [11] for our knowledge acquisition task. We briefly review some basic concepts related to it. For data representation we use Datalog which is a restriction of Prolog formalism. In Datalog a *term* is defined as a constant symbol or a variable. To distinguish between them, we write variables with an initial upper case letter, while using names beginning with lower case letters for constants. An *atom* is an  $m$ -ary *predicate* symbol followed by a bracketed  $m$ -tuple of terms. A *definite clause* is a universally quantified formula of the form  $B \leftarrow A_1, \dots, A_n (n \geq 0)$ , where  $B$  and the  $A_i$  are atoms. This formula can be read as “*B if  $A_1$  and ... and  $A_n$* ”. If  $n = 0$  a definite clause is also called a *fact*. *Ground* clauses are clauses that contain only constants as terms, no variables. A substitution  $\theta$  is a set of bindings  $\{X_1/a_1, \dots, X_m/a_m\}$  of variables  $X_i$  to terms  $a_i$ . If we substitute  $a_i$  for each occurrence of  $X_i$  in a clause  $C$ , we obtain  $C\theta$ , the instance of  $C$  by substitution  $\theta$ . If  $C\theta$  is ground, it is called a ground instance of formula  $C$ , and  $\theta$  is called a grounding substitution. The search space for first order data mining algorithms is defined by mean of  $\theta$ -subsumption relation. Clause  $C$   $\theta$ -subsumes  $D$  iff there exists a substitution  $\theta = \{X_i/V_{i_j}\}$ , where  $X_i$  ranges over the variables in  $C$  and  $V_{i_j}$  ranges over the variables and constants in  $D$ , such that  $C\theta \subseteq D$ .

A deductive Datalog database is a set of definite clauses. Often a distinction is made between the extensional database, which contains the predicates defined by means of ground facts only, and the remaining intensional part.

A Datalog (and Prolog) query is a logical expression of the form  $?-A_1, \dots, A_n$ . Submitting such a query to a Datalog database corresponds to asking the question “does a grounding substitution exist such that conjunction  $A_1$  and ... and  $A_n$  holds within the database”. The (resolution based derivation of the) answer to a given query with variables  $\{X_1, \dots, X_m\}$  binds these variables to terms  $\{a_1, \dots, a_m\}$ , such that the query succeeds if  $a_i$  is substituted for each  $X_i$ . This so-called *answering substitution* is denoted by  $\{X_1/a_1, \dots, X_m/a_m\}$ . Due to the nondeterministic nature of the computation of answers, a single query  $Q$  may result in many answering substitutions. We will refer by  $anserset(Q, \mathbf{r})$  to the set of all answering substitutions obtained by submitting query  $Q$  to a Datalog database  $\mathbf{r}$ .

In the FARMER [11] framework there is a further restriction on language called Object Identity bias [12] defined as follow:

Under *Object Identity* within a Datalog clause, terms (even variables) denoted with different symbols must be distinct (i.e., they must refer to different objects).

For example under OI assumption the Datalog clause  $C=p(X):-q(X, X), q(Y, a)$ . is an abbreviation for the Datalog<sup>OI</sup> (a logic language resulting from the application of OI to Datalog) clause  $C_{OI} = p(X) : -q(X, X), q(Y, a), X \neq Y, [X \neq a], [Y \neq a]$ . Such a restriction allows for a more efficient search in the space of solution because computing the query frequency and queries equivalence is more easy than using normal  $\theta$ -subsumption. This setting is close related to that of graph mining [13,14] that is considerably more efficient than the general query mining.

The query mining task is defined as follow:

Assume  $\mathbf{r}$  is a Datalog database,  $\mathcal{L}$  is a set of Datalog queries  $Q$  that all contain an atom *key*, and  $q(\mathbf{r}, Q)$  is true if and only if the frequency of query  $Q \in \mathcal{L}$  with respect to  $\mathbf{r}$  given *key* is at least equal to the frequency threshold specified by the user. The *frequent query discovery task* is to find the set  $Th(\mathcal{L}, \mathbf{r}, q, key)$  of frequent queries.

The frequency of a query  $Q \in \mathcal{L}$  that contain an atom *key* w.r.t database  $\mathbf{r}$  is defined as follow:

$$freq(Q, \mathbf{r}, key) = \frac{|\{\theta_k \in anserset(?-key, \mathbf{r}) \mid Q\theta_k \text{ succeeds w.r.t. } \mathbf{r}\}|}{|\{\theta_k \in anserset(?-key, \mathbf{r})\}|}$$

A frequent query  $Q$  is *closed* if for each query  $Q_1$  such that  $Q$   $\theta$ -subsumes  $Q_1$  (i.e.  $Q_1$  is more specific than  $Q$ ) the frequency of  $Q_1$  is less than the frequency of  $Q$ .

From the frequent queries it is possible to compute multi-relational association rules which are existential quantified formula in this form:  $A_1, \dots, A_k \Rightarrow A_{k+1}, \dots, A_n$ , where  $A_i$  are atoms. This formula should be read as “if query  $?-A_1, \dots, A_k$  succeeds then the extended query  $?-A_1, \dots, A_n$  succeeds also”

Each association rules can be characterized by a set of numerical value describing its importance. The most used measures for rule relevance are frequency

and confidence. Given a rule  $A_1, \dots, A_k \Rightarrow A_{k+1}, \dots, A_n$  its frequency is defined as the frequency of query  $?-A_1, \dots, A_n$  while its confidence is defined as the ratio of frequency of queries  $?-A_1, \dots, A_n$  and  $?-A_1, \dots, A_k$ . Given the “head” and the “body” of the rule then confidence tells how much reliable the rule is when the “head” holds. Multi relational association rules are not to be confused with Prolog Horn clauses which are universally quantified formulas and have a different logical meaning.

## 5 Mining Disjunctive Multirelational Association Rule for ITS

We now define a mechanism for extraction of rules from examples based on the query mining process that can be adapted smoothly to the constraint based tutor. The use case is that one of a teacher using an authoring tool for constructing an Intelligent Tutoring System in her knowledge domain. The teacher could not have competence in computer programming and/or artificial intelligence so we need mechanisms that can automate as much as possible the authoring phase.

The context is similar to that one described in [6] where the teacher must supply to the system a set of  $N$  problems along with their possible solutions. The system, at this point, applies an algorithm for extraction of rules that fit all the given examples and then shows them to the teacher for the validation phase and attribution of the relative feedback to each rule. The feedback is shown in tutoring phase when a rule is violated.

As it is shown in [15] the general architecture for knowledge base acquisition is made of three phases: ontology building for the domain representation, creation of a set of parser rules for solution parsing and construction of constraint base for tutoring phase. We will focus on automating the construction of constraint base even if there are some methods that can be used to automate the other two tasks. Solutions are supplied by the teacher in free text form or following a schema defined by the concepts and relations found in the domain of interest. We suppose that a parser is supplied by someone else and is able to recognize the concepts and relations described by means of a domain ontology. The parsed solution then follows the path described in figure [1] that represents the overall scheme of knowledge acquisition system.

The problem of finding a general rule that must be satisfied in a domain, can be formulated as an Inductive Logic Programming (ILP) [16] problem. In a general setting an ILP problem is formulated as follow:

Given a set of training *positive* and *negative examples*  $E$  and *background knowledge*  $B$ , where  $B$  is a set of *rules* or *facts*, find a hypothesis  $H$ , expressed in some concept description language  $\mathcal{L}$ , such that  $H$  is complete and consistent with respect to the background knowledge  $B$  and the examples  $E$ .

In our case the examples, background knowledge and hypothesis are expressed in Prolog language. We can view the data mining task as a subclass of ILP general

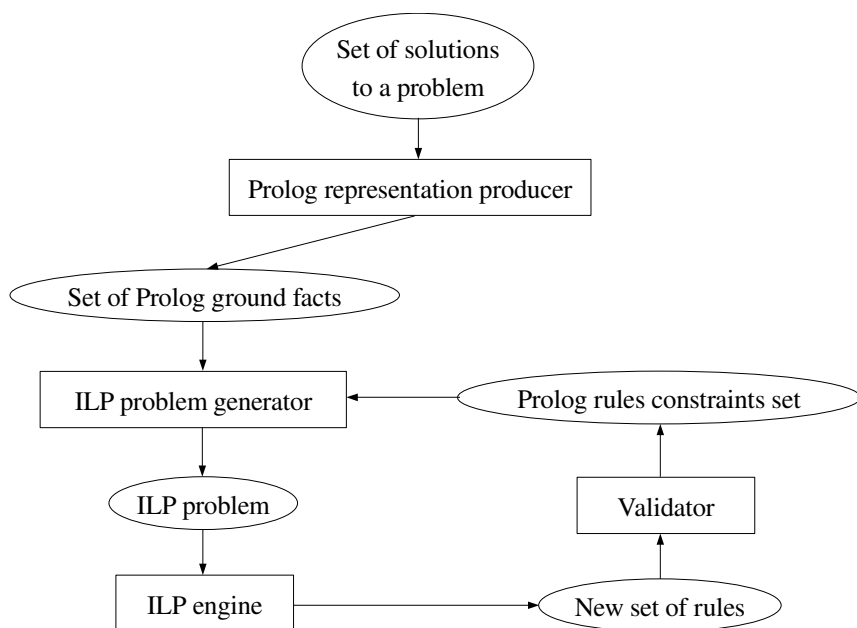


Fig. 1. Schema of knowledge acquisition module

base formulation and use also first order query mining to find suitable hypothesis that fits the given examples.

In our setting teacher gives only positive examples as a set of solutions to the same problem. Such solutions are then processed by the *Prolog representation producer* module in order to obtain a suitable representation to be processed in an ILP setting. The background knowledge is represented by the ontology opportunely transformed in Prolog terms and rules and it is used to map the solution in Prolog terms. It can be used in the second phase of rules generation to select the most interesting rules but this feature is not implemented yet.

This information are processed by another component called *ILP problem generator* that produces a set of data that can be directly processed by an external ILP engine that in our case is FARMER augmented with an algorithm to extract disjunctive association rules that we call RuleMinator. RuleMinator will produce a new set of rules representing the constraints related to the provided examples. Such set of rules must be validated before it can be added to the constraints database. The whole process can be repeated until all necessary constraints are found. Currently, constraints base has to be generated from scratch each time the teacher adds new problem but in the future we plan to change the mechanism to modify the added constraint.

The objective in the context of an ITS is finding all exact rules that is 100% confidence rules. This is needed because we must be sure that applying a rule it is valid in all the cases of resolution of the problems presented to the system.

The algorithm proceeds taking in input the examples given from the teacher and transforming them in the Prolog correspondent following the ontology scheme: every concept corresponds to an atom with single parameter while the relations correspond to an atom with two parameters. As described above the parser must be supplied for every domain unless the solution is strongly structured in parts corresponding to single concepts of the domain. For a better visualization of the rules it was adopted the strategy described in [17] in which the queries are arranged in a closed query lattice from which it is possible to extract not redundant association rules.

Once the frequent association rules are found they must be transformed in a suitable form for being used in the ITS. Until now the produced rules have been generated starting from the examples given by the teacher but for being usable by the system they have to be generated taking into account that the given examples have to be considered in a new space defined by the cartesian product of the answers to the same problem. Formally given  $n$  problems  $P_i$  having  $n_i$  solutions  $S_{i_j}$  with  $0 < j \leq n_i$  we consider the new data set  $\mathcal{D}$  defined as follow:

$$\mathcal{D} = \{S \mid \forall P_i \quad S = S_{i_j} \times S_{i_k}, \quad 0 \leq j, k \leq n_i\}$$

In this way each couple of solutions are treated as if they were respectively ideal solution and student solution obtaining consequently rules that express in the premise characteristic pertaining to the teacher solution and in the consequent the facts that must hold in order for the rule to be satisfied.

Once the rules have been generated they have to be listed to the user in comprehensible way and ordered by their importance. Since the obtained rules are all exact rules (i.e. 100% confident rules) heuristics, other than confidence, are needed in order to decide how to order rules. Currently those used are based on the length of the rule defined on the number of atoms (the more the rule is simple the more is probable it is interesting), number of disjunctions in the body of the rule (the few the number of disjunctions is the more the rule is interesting), intersection of the support of the disjunctions. The implementation details are omitted for lack of space.

The algorithm outline is shown in figure 2

1. Transform given example in input files for FARMER
2. Find all frequent query in given examples
3. Build the closed query lattice
4. Generate the disjunctive association rules
5. Transform the association rules in the new space of queries
6. Order the rules based on predefined heuristics

**Fig. 2.** Algorithm to mine disjunctive association rules in the ITS settings

### 5.1 Test for a Simple Algebra Equation Tutor

We test the algorithm with a problem on simple domain to see if the rules founded by the system are plausible. The ontology in figure 3 shows a partial representation of domain concepts and relations in the domain of first grade algebra equation.

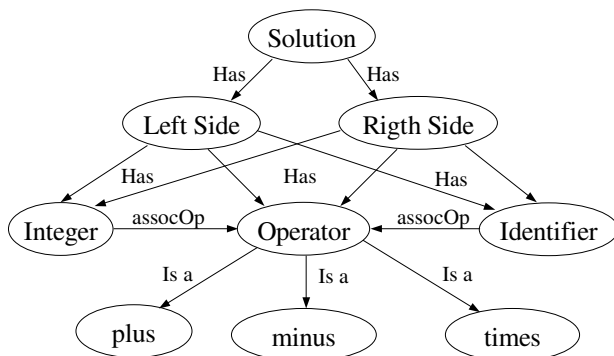


Fig. 3. Partial ontology for equation of first order

Suppose we have a problem which allows the following solutions:

- 3\*x-4=2;
- 3\*x-2=4;
- 3\*x=4+2;

We suppose to write the solution without simplification for now because of the difficult of FARMER to include intensional knowledge in the mining process. The resulting representation in Prolog fact for the first equation is:

```

solution(V2N0),has(V2N0,V2N1),lhs(V2N1),has(V2N0,V2N2),rhs(V2N2),
has(V2N1,V2N3),times(V2N3),has(V2N1,V2N4),identifier(V2N4),
has(V2N1,V2N5),integer(V2N5),has(V2N2,V2N6),integer(V2N6),
hasValue(V2N5,3),assocOp(V2N3,V2N4),assocOp(V2N5,V2N3),
.....
    
```

After running the system on this simple set of solution it presents the following rules as the most important (we report only a few to show that system found a set of plausible rules):

```

RULE 1
IF IS solution(V2N0),has(V2N0,V2N1),lhs(V2N1)
THAN SS solution(V2N0),has(V2N0,V2N1),lhs(V2N1)

RULE 2
IF IS rhs(V2N1),has(V2N1,V2N2),integer(V2N2)
THAN SS rhs(V2N1),has(V2N1,V2N2),integer(V2N2)
    
```

## RULE 3

```

IF IS solution(V2N0),has(V2N0,V2N1),lhs(V2N1),has(V2N1,V2N2),
times(V2N2),has(V2N1,V2N3),identifier(V2N3),assocOp(V2N2,V2N3)
AND SS solution(V2N0),has(V2N0,V2N1),lhs(V2N1),has(V2N1,V2N2),
times(V2N2),has(V2N1,V2N3),identifier(V2N3),
THAN SS assocOp(V2N2,V2N3)

```

## RULE 4

```

IF IS solution(V2N0),has(V2N0,V2N1),lhs(V2N1),
has(V2N1,V2N2),minus(V2N2)
THAN SS solution(V2N0),has(V2N0,V2N1),lhs(V2N1),
has(V2N1,V2N2),minus(V2N2)
OR SS solution(V2N0),has(V2N0,V2N1),rhs(V2N1),
has(V2N1,V2N2),plus(V2N2)

```

## RULE 5

```

IF IS solution(V2N0),has(V2N0,V2N1),rhs(V2N1),
has(V2N1,V2N2),integer(V2N2),hasValue(V2N2,X)
THAN SS solution(V2N0),has(V2N0,V2N1),rhs(V2N1),
has(V2N1,V2N2),integer(V2N2),hasValue(V2N2,X)
OR SS solution(V2N0),has(V2N0,V2N1),lhs(V2N1),
has(V2N1,V2N2),integer(V2N2),hasValue(V2N2,X)

```

Rule 1 states that if an ideal solution has a left hand side then the student solution must have a left hand side. Rule 2 states that if right hand side of equation has an integer then student solution must have an integer on the right hand side. The second rule is plausible, even if not always true in general, because it holds in all the given examples. Rule 3 states that if there are an integer and a times operator related by an “assocOp” relation in the ideal solution and there are also an integer and a times operator in the student solution they have to be related by “assocOp” relation too. Rule 4 states that if in the left hand side of equation there is a minus operator then the student solution must have a minus on the left side or a plus on the right side. The rule 5 states that if in the right side of the equation there is an integer with value  $X$  then the student solution must have an integer with the same value in one of the side.

## 6 Conclusion and Further Work

We have presented a framework for knowledge acquisition in an Intelligent Tutoring System based on first order data mining that allows to acquire disjunctive multi-relational association rules. This method can be applied in constraint based tutor and with some modification to the initial representation to pseudo-cognitive tutor. The main advantage respect to other framework is that this approach is based on a strong theoretical bases of first order data mining and can improve its efficacy as the research on this fields goes on. Once we have a representation of solution for an ILP engine, we do not need to develop an



ad hoc mechanism in order to find the constraint rules. Currently we suppose that both a parser and ontology are given by someone else but there are some mechanisms that can partially automate the acquisition process of ontology and parser rules. We plan to implement a tutor by means of this framework for “C” programming language and try it out in a first year academic course of computer science faculty.

## References

1. Koedinger, K.R., Anderson, J.R.: Intelligent tutoring goes to the big city. *International Journal of Artificial Intelligence in Education* 8, 30–43 (1997)
2. Act-r a cognitive theory about human cognition, <http://act-r.psy.cmu.edu/>
3. Anderson, J.R., Reiser, B.: The lisp tutor. *Byte* 10(4), 159–175 (1985)
4. Mitrovic, A., Mayo, M., Suraweera, P., Martin, B.: Constraint-based tutors: A success story. In: IEA/AIE, pp. 931–940 (2001)
5. Martin, B., Mitrovic, A.: Wetax: A web-based authoring system for constraint-based its. In: AH, Malaga, pp. 543–546 (2002)
6. Suraweera, P., Mitrovic, A., Martin, B.: The role of domain ontology in knowledge acquisition for itss. In: *Intelligent Tutoring Systems, Brazil*, pp. 207–216 (September 2004)
7. Suraweera, P., Mitrovic, A., Martin, B.: A knowledge acquisition system for constraint-based intelligent tutoring systems. In: *Artificial Intelligence in Education, Amsterdam*, pp. 638–646 (2005)
8. Koedinger, K.R., Aleven, V., Heffernan, N.T.: Toward a rapid development environment for cognitive tutors. In: *12th Annual Conference on Behavior Representation in Modeling and Simulation. Simulation Interoperability Standards Organization* (2003)
9. Koedinger, K.R., Aleven, V., Heffernan, N.T., McLaren, B.M., Hockenberry, M.: Opening the door to non-programmers: Authoring intelligent tutor behavior by demonstration. In: *Intelligent Tutoring Systems, Brazil*, pp. 162–174 (2004)
10. Suraweera, P., Mitrovic, A., Martin, B.: The use of ontologies in its domain knowledge authoring. In: *Workshop on Applications of Semantic Web for E-learning, Brazil, Maceio*, pp. 41–49 (September 2004)
11. Nijssen, S., Kok, J.N.: Efficient frequent query discovery in farmer. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) *PKDD 2003. LNCS (LNAI)*, vol. 2838, pp. 350–362. Springer, Heidelberg (2003)
12. Lisi, F.A., Ferilli, S., Fanizzi, N.: Object identity as search bias for pattern spaces. In: *ECAI*, pp. 375–379 (2002)
13. Nijssen, S., Kok, J.N.: The gaston tool for frequent subgraph mining. In: *International Workshop on Graph-Based Tools, Rome, Italy*, pp. 77–87. Elsevier, Amsterdam (2004)
14. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: *ICDM 2002*, pp. 721–724 (2002)
15. Riccucci, S., Carbonaro, A., Casadei, G.: An architecture for knowledge management in intelligent tutoring system. In: *CELDA, Porto, Portugal*, pp. 473–476 (December 2005)
16. Lavrač, N., Džeroski, S.: *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, New York (1994)
17. Stumme, G.: Iceberg query lattices for datalog. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) *ICCS 2004. LNCS (LNAI)*, vol. 3127, pp. 109–125. Springer, Heidelberg (2004)

# Features Selection Through FS-Testors in Case-Based Systems of Teaching-Learning

Natalia Martínez, Maikel León, and Zenaida García

Department of Computer Science, Central University of Las Villas  
Highway to Camajuaní km 5 ½, Santa Clara, Cuba  
Phone: (53) (42) 281515  
{natalia,mle,zgarcia}@uclv.edu.cu

**Abstract.** The development of intelligents teaching-learning systems depends, on one hand, of the pedagogical paradigms and, on the other hand, of the available technologies to implement these paradigms in computers. The field of the Intelligent Teaching-Learning Systems is characterized by the application of Artificial Intelligence techniques, to the development of the teaching-learning process assisted by computers, where the term "intelligent" is associated to the student's aptitude to dynamically acclimatize to the teaching process by carrying out an individual learning. The case-based reasoning is an Artificial Intelligence technique that performs their reasoning process based on previously solved cases, stored in case-bases. In this article we propose a new case-based approach with foundations on fuzzy pattern recognition to help elaborate intelligents teaching-learning systems, using the *FS*-testor theory, based on a combination of typical testor theory with the fuzzy sets, assures the efficient access and retrieval of cases.

## 1 Introduction

Imprecision permeates our understanding of the real world. It is related to the information which is not certain, complete or precise. The purpose of Information Systems is to model the real world. Thus, to build useful Information Systems, it is necessary to learn how to represent imperfect information and to reason with it. Up to now, the question of imprecision has not been fully studied in case-based systems due to the fact that flexibility in the representation of imprecision information has affected efficiency while searching for new solutions. Although pattern recognition and artificial intelligence branches have usually been confronted and have developed in isolation from each other, the recent tendency is to try to combine their tools. Their common difficulties and challenges arise from the common classification task. The main emphasis however should be put not on the common tools but on the integration of these different approaches. Problems which appear cumbersome for the physician are naturally difficult to interpret in terms of artificial intelligence (the well known knowledge acquisition bottleneck). The intelligent teaching-learning Systems constitute a group of teaching applications that promote an individual and flexible learning based on the knowledge and the student's. Up to now these systems have demonstrated their effectiveness in diverse domains. However their construction

implies a complex and intense work of engineering of the knowledge, what impedes a more general and taken advantage of use. To eliminate these obstacles thinks about the objective of building author's tools that facilitate the construction from to people non experts, in particular to the own instructors that dominate a certain teaching matter [1]. Humans seem to pick up just the relevant information and ignore the irrelevant. This ability may account for the speed with which reasoning is performed in everyday situations. In this paper, relevance is viewed as a notion that can be used to reduce the computational cost of reasoning.

The Case-based reasoning is an Artificial Intelligence technique that allows taking advantage of the experience accumulated in the solution of our problems. With this technique cases are stored with the solution that has been given previously and when a new problem with this information is presented or accumulated experience is an employee to solve it. Organizing and indexing cases in memory is a fundamental part of case-based reasoning that involves learning and reasoning processes. This problem can be divided into two parts. The first one is the selection of the features of the cases that can be used to index and retrieve the cases. The second one is the organization of the case memory so that the case retrieval process is efficient and accurate. In this paper we present a case-based approach for teaching-learning systems independently from application domain. The model proposed contains a module, capable of building a representation of the user objectives and characteristics that allows the system to personalize the teaching-learning interaction. This modulates allows to treat information imprecise, as categories of excellent, well, regulate, bad, etc, in terms of fuzzy set. A distinguishing feature of the model is the use of FS-testor theory taking as theoretical mark typical testor theory combined with the fuzzy sets in the selection of relevant features and cases. The proposed model was implemented in the HESEI computational system, successfully applied in the making of decisions in teaching-learning systems by non-expert teachers in the field of informatics in domains where experts are, starting from a conceptual map and several questionnaires designed methodologically to capture the student cognitive state in different moments.

## 2 Information Systems

A table can represent a data set where each row represents, for instance, an object, a case, or an event. Every column represents an attribute that is a variable, an observation, or a property that can be measured for each object. This table, more formally called an Information System, is a pair  $A = (U, A)$  where  $U$  is a none-empty finite set of objects called the universe and  $A$  is a none-empty finite set of attributes such that  $a : U \rightarrow V_\alpha$  for every  $a \in A$ . The set  $V_\alpha$  is called the value set of  $\alpha$ . The choice of attributes is subjective and reflects our intuition about factors influencing the classification of objects in questions.

### 2.1 Imprecision or Ambiguities in Information Systems

In a fuzzy information system, things may be completely true, completely false, or anything in common. Ambiguities abound in fuzzy systems, and contradictions are

frequently encountered; fuzzy systems provide structured ways of handling uncertainties, ambiguities, and contradictions, none of which are ordinarily encountered in conventional computer programming. Fuzzy systems also employ some special terms and concepts: fuzzy sets, fuzzy numbers, and membership functions.

### 2.2 Case-Based Systems as Information Systems

In many cases the outcome of the Information System is provided by an additional attribute called decision. Information Systems of this kind are called Decision Systems [2]. A Decision System is any Information System of the type  $A = (U, A \cup \{d\})$ , where  $d \notin A$  is the decision attribute. There are two types of case-based systems: interpretative and problem solver. A case-based system problem solver is a decision system where the universe represents the case set and where the attributes  $A$  are called predictor features and decision  $d$  is called the objective feature. The fundamental components of a Case-Based System are: the Case-Base, the Recovery Module and the Adaptation Module.

**Case-Base:** Is described starting from the values that are assigned to the predictor features and to the objective ones. A case that has  $n$  predictor features and an objective feature is described in the following way:

$O_t(x_1(O_t), \dots, x_i(O_t), \dots, x_n(O_t), y(O_t))$  where  $x_i(O_t)$ : Value of the predictor feature  $i=1, n$  for the case  $O_t$   $t=1, m$  of the case-base and  $y(O_t)$ : Value of the objective feature for the case  $O_t$   $t=1, m$  of the case-base. The case-base is made up by a group of cases that can be represented through a decision table, in which, the columns are labeled by variables representing the predictor and objective features and the rows represent the cases. Table 1 shows a case-base represented as a decision table.

**Table 1.** A Case-base represented as a decision table

Case	$x_1$	...	$x_n$	$y$
O1	$x_1(O_1)$	...		$y(O_1)$
...	...	...	...	
Om	$x_1(O_m)$	...	$x_n(O_m)$	$y(O_m)$

A new problem is represented starting from the predictor features in the following way:  $O_0(x_1(O_0), \dots, x_i(O_0), \dots, x_n(O_0))$ . The main goal of the Case-Based Systems (Problem Solvers) is to solve the new problem from the cases stored in the case-base.

**Recovery Module:** To solve the new problem the most similar cases are obtained from the recovery module. In many practical problems to increase the efficiency of this process, it is necessary to reduce the set of cases. The two key aspects of this

phase are the access algorithm to cases and the similarity measure among cases. To determine how similar a case is to another, several techniques have been developed. The simplest one consists of the use of a group of heuristics that allow us to determine which characteristics have greater relevance in order to formulate a similarity function involving the similarity among each feature with relevance in mind [3]. A mathematical model of this technique is the Near-Neighbor Similarity Function:

$$\beta(O_0, O_t) = \frac{\sum_{i=1}^n p_i \delta_i(O_0, O_t)}{\sum_{i=1}^n p_i}$$

where:  $P_i$ : Weight or relevance of the feature  $i$  and  $\delta_i(O_0, O_t)$ : Comparison function between the cases  $O_0$  and  $O_t$  according to the feature  $i$ . For example:

$\delta_i(O_0, O_t) = 1 - \frac{ x_i(O_0) - x_i(O_t) }{ x_i(O_0) + x_i(O_t) }$	$\delta_i(O_0, O_t) = \begin{cases} 1 & \text{if } x_i(O_0) = x_i(O_t) \\ 0 & \text{in other case} \end{cases}$
--	---

On the other hand, the access algorithm to the cases should be quick and efficient. This depends on the organization techniques used. Among them we can mention:

- Exact access: The case is found if it fits exactly with the new case.
- Hierarchical access: Cases are stored in a tree structure. Their search is done in each node until there is no possibility to advance. If a leaf is found, the corresponding cases are shown. If a node is arrived at, all the cases that are derived are shown.

**Adaptation Module:** The adaptation can either be performed by the user or automatically executed by the system. If the user does the adaptation, the system just carries out the search of similar cases. If the adaptation is performed by the system, then it should contain some knowledge, such as formulas or rules. The developed techniques for the adaptation in an automatic way have generally reported good results. In [3] the following methods of adaptation are mentioned:

- Reinstantiation: In this type of technique the mark or context of previous situations is used but with new arguments.
- Adjustment of parameters: The parameters of the recovered cases are adjusted according to the difference between their description and the description of the new problem.
- Local search: Searches inside semantic hierarchies are carried out and the problem is solved by analogy among other methods.

### 3 The FS-Testor Theory

If membership degrees are supposed to be numerical then there should be some operational definition of these numbers. There are basically three kinds of quantities

that can help measuring fuzzy set membership: distance, frequency and cost. Distance is obvious when a membership function is interpreted in terms of similarity [4]. The concept of *FS*-testor for any type of similarity function, with the characteristic that the comparison among 1-uplos of membership to the classes is carried out with an expression boolean like in the case of the *g*-testor, given a couple of objects, to the effects of the testor definition, both are in oneself class or are in different classes. Another formulation exists for the case in that this comparison is not necessarily this way, the one that can be applied to the general case of diffuse classes. The definition group *FS*-differentiate preserve the essence of the definition of classic testor of Zhuravlev [5], is a subset of features that maintains or it improves the capacity differentiate of a given reference group.

Given a subset *R* and a reference group, you can determine if the first one is *FS*-testor with regard to the second and a group of parameters. The subset completes the property settled down in the definition or it doesn't complete it. As a result of the application of the previous approach, a family of classic or "hard" *FS*-testor is obtained. However, it is interesting to define the diffuse family of that *FS*-testor in the following way: all the subsets that complete the definition belong to the family with membership degree 1, the rest of the subsets belongs with a membership degree in the open interval  $[0, 1)$ . The membership degree of a subset will be bigger in the measure in that comes closer more to the execution of the property expressed in the definition [6]. The diffuse family of *FS*-testor has for objective to differ to all the subsets of features that are not *FS*-testor, a magnitude that informs settling down how it fences or it is not a subset of completing is also part of the definition of *FS*-testor.

## 4 Fuzzy Case-Based Model

Despite the recent activity, and the associated progress, in methods for selecting relevant features and cases, there remain many directions in which case-based approach can improve its study of these important problems. Here we outline some research challenges for the theoretical and empirical learning communities. The developed model is based on a representation structure based on category and exemplar and it uses some basic concepts of the theory of the fuzzy set and *FS*-testors [6] for the calculation of membership degree. Fuzzy logic is a computational paradigm that provides a mathematical tool for representing and manipulating information in a way that resembles human communication and reasoning processes. It is based on the assumption that, in contrast to Boolean logic, a statement can be partially true (or false).

Fuzzy modeling is the task of identifying the parameters of a fuzzy inference system so that a desired behavior is attained. Note that the fuzzy-modeling process has to deal with an important trade-off between the accuracy and the interpretability of the model. In other words, the model is expected to provide high numeric precision while incurring as a little loss of linguistic descriptive power as possible. With the direct approach a fuzzy model is constructed using knowledge from a human expert. This task becomes difficult when the available knowledge is incomplete or when the problem space is very large, thus motivating the use of automatic approaches to fuzzy modeling. Formally, a fuzzy set *A* defined over a set of values *X* is a pair  $(x, \phi_A(x))$

where  $\varphi_A(x)$  is called the membership function of set  $A$ . The membership function assigns to each element of  $X$  a membership degree to  $A$  in the interval  $[0, 1]$ .  $X$  is called “universe” and it can be a discrete or continuous space.

The developed model is designed in a representation structure based on categories and exemplars and it uses some basic concepts of the theory of the diffuse groups [7] and the typical testors for the selection of the relevant features and its importance. The PROTOS system [8] proposes an alternative way to organize cases in a case memory. The case memory is embedded in a network structure of categories, cases and index pointers; each case is associated with a category. An index may point to a case or a category. Within this memory organization the categories are interlinked within a semantic network, which also contains the features and intermediate states. This network represents a background of a general domain knowledge, which enables explanatory support to some of the case-based reasoning task.

#### 4.1 Representation of Case-Base

From the case-based reasoning perspective, membership degree problems are described in the following way:

- A group of features that describe the cases (predictor features).
- A possible decision to be made (objective feature).
- The membership degree present in the predictor features.
- The membership degree present in the objective feature.

Consequently, the case-base is made up of a group of predictor features and of an objective feature adopting a form of (value, membership degree), where the values are discrete with a finite cardinality. A new problem is described in terms of the values that the predictor features acquire in the new situation, where the agent has to make a decision and the system has to infer what decision to make starting from the experience accumulated in the case-base. Then the characteristics of the proposed model are the following:

1. It assures an efficient representation of the case-base.
2. It gives a way to calculate the membership degree present in the cases stored in the case-base, because the case-base is usually created starting from sources not registered explicitly, that is, for each value  $x_i(O_t)$  given to the predictor feature  $i$  ( $i=1, n$ ) in the case  $O_t$  ( $t=1, m$ ) we should find a measure  $\mu_i(O_t)$  that denotes the membership of this value. Similarly, for each decision  $y(O_t)$  given to the objective feature in the case  $O_t$  ( $t=1, m$ ), a measure  $\nu(O_t)$  denoting the membership of this value should be searched.
3. It develops a recovery procedure for the selection of the cases most similar to the new problem starting from the descriptions outlined in the case and the situation.
4. It develops a process of adaptation that determines what decision to make in the new problem on the grounds of the decisions of the recovered cases and membership degree.

### 4.2 Using FS-Testor Theory to Calculate the Relevant Features

The intelligent teaching-learning system is structured in topics (predictor’s features) which take its value through a questionnaire of  $n$  questions. The values that can take the topics are not necessarily disjoint, for what each topic is formed by the even value-grade of ownership. For each value of each topic exists an exemplar that was obtained from an experts' approach. As it was already exposed previously, the model is conceived for non expert in the computer field for what becomes necessary to analyze the possibility to reduce the representation space, for is applied it and algorithm 1 to each one of the topics to select the relevant questions, that is to say the useful ones in the evaluation of the same ones.

**Algorithm 1:** Selection of relevant features using the FS-testors.

For all the topics do:

Be the learning matrix  $MA_{2^n \times n}$ ,  $n$  is the number of questions with which the topic is evaluated, divided in  $\eta$  class and  $\eta$  is the cardinality of the set of values that this predictor feature takes.

**Step 1:** Using an external scale algorithm LEX [9] to calculate the conjunct  $S$  of all typical testor. Formed the conjunct  $R^*$  with all the features (questions) that form part of at least a typical testor.

**Step 2:** Using a training matrix  $MA$  (table2), and  $R^*$  a subset of  $R$ . Function of comparison of 1-uplos of membership to the classes [4]:

$$V: [0, 1] \times [0, 1] \rightarrow v' \quad v(\bar{\alpha}(O_i), \bar{\alpha}(O_j)) = \begin{cases} 1 & \text{si } \bar{\alpha}(O_i) = \bar{\alpha}(O_j) \\ 0 & \text{si } \bar{\alpha}(O_i) \neq \bar{\alpha}(O_j) \end{cases} \quad (1)$$

**Table 2.** Learning Matrix

	$x_1, \dots, x_n$	$\alpha_1, \dots, \alpha_\theta$
$O_1$	$x_1(O_1), \dots, x_n(O_1)$	$\alpha_1(O_1), \dots, \alpha_\theta(O_1)$
$\vdots$	$\vdots$	$\vdots$
$O_m$	$x_1(O_m), \dots, x_n(O_m)$	$\alpha_1(O_m), \dots, \alpha_\theta(O_m)$

where

$$\alpha_\theta(O_m) = \frac{\sum_{i=1}^n \delta_i(O_m, O') \sum_{i=1}^n i \delta_i(O_m, O')}{\frac{n^2(n+1)}{2}} \quad (2)$$

$$\delta_i(O_m, O') = \begin{cases} 1 & \text{if } x_i(O_m) = x_i(O') \\ 0 & \text{in other case} \end{cases} \quad (3)$$

and  $D=1$ , it is the subs  $y_\theta$  t of  $V'$  where two 1-uplos of membership is considered similar.  $R^*=S$ , conjunct of all typical testor.



1. Calculate all *FS*-differentiate according to [4]:

$T \subseteq R$  ( $T$  diffuse subset of  $R$ ) it is a group *FS*-differentiate of features with regard to,  $D'$ ,  $R^*$  and of  $MA$  if  $\forall O_i, O_j \in MA [v(\overline{\alpha}(O_i), \overline{\alpha}(O_j)) \notin D'] \Rightarrow [\beta(I/T(O_i), I/T(O_j)) \preceq \beta(I/R^*(O_i), I/R^*(O_j))]$

2. Calculate the *FS*- characterized starting from the group of the *FS*-differentiate according to [4]:

$T \subseteq R$  is a combined *FS*- characterized of features with regard to,  $D'$ ,  $R^*$   $MA$  if and only if  $\forall O_i, O_j \in MA [v(\overline{\alpha}(O_i), \overline{\alpha}(O_j)) \in D'] \Rightarrow [\beta(I/R^*(O_i), I/R^*(O_j)) \preceq \beta(I/T(O_i), I/T(O_j))]$ .

3. Obtained the conjunct of *FS*-testor according to [4]:

$T \subseteq R$  is a *FS*-testor with regard to,  $D'$ ,  $R^*$  and of  $MA$  if and only if it is at the same time a group *FS*-differentiate and *FS*- characterized of  $MA$  with regard to the same parameters.

**Step 3:** Calculate the diffuse family of *FS*-testor according to [4]:

Be  $OD = \{(O_i, O_j) \in MA / v(\overline{\alpha}(O_i), \overline{\alpha}(O_j)) \notin D'\}$  y  $C(OD)$  the complement of the conjunct  $OD$ , that is to say,  $C(OD) = \{(O_i, O_j) \in MA / v(\overline{\alpha}(O_i), \overline{\alpha}(O_j)) \in D'\}$ .

Be  $T$  and  $R^*$  subsets of  $R$ . Are  $S^{R^*}(T)$  and  $D^{R^*}(T)$  the following groups:

$$S^{R^*}(T) = \{(O_i, O_j) \in OD, \beta(I_{/T}(O_i), I_{/T}(O_j)) \succ \beta(I_{/R^*}(O_i), I_{/R^*}(O_j))\} \tag{4}$$

$$D^{R^*}(T) = \{(O_i, O_j) \in C(OD), \beta(I_{/T}(O_i), I_{/T}(O_j)) \prec \beta(I_{/R^*}(O_i), I_{/R^*}(O_j))\} \tag{5}$$

Be  $\zeta$  a diffuse family of that *FS*-testor:

$$\zeta = \{Tp \mid \mu\zeta(Tp): Tp \subseteq R\} \tag{6}$$

$$\mu_\zeta(T_p) = \delta_1 \eta_1(T_p) + \delta_2 \eta_2(T_p) + \delta_3 \eta_3(T_p) \tag{7}$$

$$\eta_1(T_p) = 1 - \frac{|S^{R^*}(T_p) \cup D^{R^*}(T_p)|}{|OD \cup C(OD)|} \tag{8}$$

$$\eta_2(T_p) = \begin{cases} 1 & sS^{R^*}(T_p) = \emptyset \\ 1 - \frac{1}{|S^{R^*}(T_p)|} \sum_{(O_i, O_j) \in S^{R^*}(T_p)} [\beta(I_{/T_p}(O_i), I_{/T_p}(O_j)) - \beta(I_{/R^*}(O_i), I_{/R^*}(O_j))] & sS^{R^*}(T_p) \neq \emptyset \end{cases} \tag{9}$$

$$\eta_3(T_p) = \begin{cases} 1 & siD^{R^*}(T_p) = \emptyset \\ 1 - \frac{1}{|D^{R^*}(T_p)|} \sum_{(O_i, O_j) \in D^{R^*}(T_p)} [\beta(I_{/R^*}(O_i), I_{/R^*}(O_j)) - \beta(I_{/T_p}(O_i), I_{/T_p}(O_j))] & siD^{R^*}(T_p) \neq \emptyset \end{cases} \tag{10}$$

and  $\delta_1, \delta_2$  and  $\delta_3$  they are coefficients pondered such that  $0 \leq \delta_i \leq 1, i = 1, 2, 3$  and  $\delta_1 + \delta_2 + \delta_3 = 1$ .

**Step 4:** Denoted  $\epsilon_i$  as importance of the feature  $i$  and calculate this way:

$$F_j(p) = \frac{\sum_{i=1}^n \mu_{\zeta}(Ti)}{\sum_{p=1}^{|FS|} \mu_{\zeta}(T_p)} \qquad L_j(p) = \frac{\sum_{j=1}^n \frac{1}{|t_j|}}{|FS|} \qquad (11, 12)$$

and  $n$  is the number of times where feature  $i$  appears,  $|t_j|$  it is the longitude of the testor  $j$  and  $|FS|$  is the cardinality of the  $FS$ -testors set.

$$\epsilon_j(p) = \alpha F_j(p) + \beta L_j(p) \qquad (13)$$

$\alpha > 0$ ,  $\beta > 0$  and  $\alpha + \beta = 1$ .  $\alpha$  y  $\beta$  are two parameters that ponder the participation or influence of  $F(p)$  and  $L(p)$  respectively in  $\epsilon$ ; this is, the importance that is granted for the relevance to the appearance frequency combined with the membership degree of the one  $FS$ -testor to the diffuse family and the longitude of those  $FS$ -testors. In the proposed model  $\alpha = 0.5$  and  $\beta = 0.5$ .

**Step 5:** Select  $FS$ -testor minimum.

1. Be  $S_m$ , conjunct of all the  $FS$ -testors the minimum cardinality.
2. For each  $FS$ -testor of  $S_m$  calculated:

$$\psi_i(t) = \sum_{j \in |t|} \epsilon_j(t)$$

3. Select the  $FS$ -testor minimal of bigger  $\psi_i(t)$

This algorithm is applied for the selection of the relevant questions for the evaluation of each topic, with its respective importance. As well as for the selection of the relevant features (topics) and the importance associated to the same ones. The algorithm is only applied in the phase of elaboration of the teaching-learning system achieving the information that is stored in the case base in an efficient way.

### 4.3 Using FS-Testor Theory in the Obtaining of a New Case

**Algorithm 2:** Obtaining of the New Case (Value and membership grade associated of each feature predictor)

For each topic to apply:

**Sp1:** Apply questionnaire associated to the topic and to obtain the result  $v'$  (vector of 0 and 1 of longitude  $n$ ).

**Sp2:** Obtain  $\mu_i = \beta(v', E_i)$ , where  $\beta(v', E_i)$  is obtained applying (1) and (2) y  $E_i$  is the exemplar of the value  $i$  whose membership grade is 1, where  $i = 1..n$  and  $n$  is the cardinality of the group of values that can take the topic. Using for the obtaining of  $\mu_i$  the outstanding questions and the importance associated to the same ones as a result of applying the Algorithm 1.

**Sp3:** Selected  $\mu_i$  bigger and to assign to the topic the even value  $i$  y  $\mu_i$  (value, membership grade).

### 4.4 Most Similar Cases Recovery

Having calculated the membership degree of all the values of the predictor features as well as the objective feature and the membership degree of the new case values, we are able to carry out the reasoning processes of recovery and adaptation. The recovery process involves two procedures: access and retrieval. The similarity function using is an adaptation of the proposal in [3].

**Access Algorithm:** During the access phase, potential cases for the recovering process are selected, using the hierarchical structure of the case representation. This structure allows us to reduce the number of cases to consider during the recovering phase since only those cases with values similar to those of the new problem are considered.

**Algorithm 3: Recovery under diffuse conditions**

Input: Problem to solve:  $O_0$ ; strong examples ( $O_F$ ) and weak ( $O_D$ ) of each category, obtained by experts' approach.

Output: Set of cases to recover: S

**R1:** For each  $O_F$  and  $O_D$  of all the categories to determine:

$$\beta(O_0, O_i) = \frac{\sum_{i=1}^n p_i \cdot \delta'_i(x_i(O_i), x_i(O_0))}{\sum_{i=1}^n p_i} \tag{14}$$

$\delta'_i(x_i(O_i), x_i(O_0)) = \delta_i(x_i(O_i), x_i(O_0))(1 - |\mu_i(O_0) - \mu_i(O_i)|)$  where  $n$ : Number of predictor features and  $p_i$ : Weight or relevance of the feature  $i$ .

**R2:** Calculate the membership grade of  $O_0$  to each category.

If  $\beta(O_o, O_F) = 1$  then  $\mu_C(O_o) = \beta(O_o, O_F)$  else  $\mu_C(O_o) = g(\beta(O_o, O_F), (\beta(O_o, O_D)))$

Finally, we calculate the membership degree of  $O_0$  for each categories using one of the combination functions used in literature to combine uncertainties. These functions are known as co-t norms, for example the sum probabilistic generalized according to [10]:  $g(x, y) = \omega_1 x + \omega_2 y - \omega x \cdot y$

**R3:** Selection of the cases more similar to  $O_0$  using (3), inside the selected category (of bigger  $\mu_C(O_o)$ ).

### 4.5 Decision Determination

Once the most similar cases have been selected, it is possible for them to suggest different decisions. Therefore, it will be necessary to determine which decision to

make. This selection is carried out considering the degree of similarity calculated in the previous procedure and the membership of the decision of the recovered cases.

$\mu(\beta(O_0, O_t), \eta_j(O_t))$ . From the point of view of the Decision-Making Theory, the result of this function could be considered as the expected utility of the case. This allows the agent to choose one case to make a decision.

**Algorithm 4**

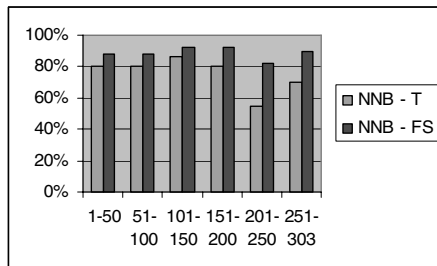
A1:  $\forall O_t \in S$  do:

$\mu(O_t) = \gamma \cdot \beta(O_0, O_t) + (1 - \gamma) \cdot \eta_1(O_t)$  where  $\alpha$  is a parameter that is selected according to the experts' criteria. If  $\alpha$  tends to 1, it means that more importance is given to the similarity between the new problem and the recovered case than to the certainty of the solution of that case.

A2: If  $\mu(O_t)$  is maximum, select case  $O_t$  and take its decision for solving the new problem.

**4.6 Results**

The model described has been implemented through HESEI (Tool to elaborate intelligent teaching-learning systems). The analysis of the efficiency of the system is carried out having in consideration the results obtained in the solution of new problems, using the similarity function of proposed (3) and the similarity function of the Nearest Neighbor in its classic form, using the adaptation method by reinstantiation. To identify these two processes, the first one is called NNB FS (New Model of the Nearest Neighbor) and the second one NNB T (Traditional Model of the Nearest Neighbor). Each experiment randomly partitioned the data into a 70% training set and a 30% testing set. It was repeated 6 times using NNB T and NNB FS algorithms. Figure 1 shows the results. With little or no domain-specific background knowledge it could be observed that the knowledge organization and the retrieval and adaptation mechanisms with membership degree handling produce meaningful results.



**Fig. 1.** Percentage of well-classified cases using the NNB T and NNB FS algorithms

**5 Conclusions**

This research focuses the problem of decision-making under ambiguities or imprecision of information in a Case-Based System pursuing a new structure for the

organization of a case-base capable of combining the use of category and exemplar structure with the *FS*-testors theory. Besides allowing a more efficient access to cases, the structure proposed makes the calculation of the existing membership degree in their values easier. The algorithms for the calculation and handling of the ambiguities or imprecision in the recovery and adaptation modules were implemented in the HESEI Computational System, successfully applied in the decisions making in teaching-learning systems.

## Referentes

1. García, Z.: Investigación y Elaboración de Sistemas de Enseñanza Inteligentes. Universidad Central de Las Villas, Santa Clara, Cuba (1993)
2. Gutiérrez, I., Bello, R.: Determination and Handling of Uncertainty in Case-Base Systems. In: Proceedings of the 7th Joint International Iberoamerican Conference. Conference on Artificial Intelligence, 15th Brazilian Conference on AI (2000)
3. Gutierrez, I., Bello, R.: A decision case-based system that reasons in uncertainty conditions. Springer, Heidelberg (2002)
4. Lazo Cortés, M.: Una generalización del concepto de testor, Aportaciones Matemáticas, Serie comunicaciones 14, IMATE-UNAM (1994)
5. Shulcloper, J.: Introducción al Reconocimiento de Patrones (Enfoque Lógico-Combinatorio). Serie Verde No. 51, CINVESTAV-IPN, México (1995)
6. Alba, E.: El concepto de FS-testor: una solución para un problema de incompatibilidad. Revista Ciencias Matemáticas (2002)
7. Didier, D.: The three semantics of fuzzy sets. Fuzzy Set and Systems (1997)
8. Bareiss, D.: Exemplar-based knowledge acquisition: A unified approach to concept representation, classification, and learning. Academic Press, Boston (1989)
9. Pons, A.: Lex: un nuevo algoritmo para el cálculo de los testores típicos. Revista ciencias matemáticas 21(1) (2003)
10. Hatzilygeroudis, I.: An Expert System with Certainty Factors for Predicting Student Success. Springer, Heidelberg (2004)

# Heuristic Optimization Methods for Generating Test from a Question Bank

Mehmet Yildirim

Electronics and Computer Education Department, University of Kocaeli, 41380,  
Kocaeli, Turkey  
myildirim@kou.edu.tr

**Abstract.** In this study, heuristic optimization methods which are genetic algorithm (GA), simulated annealing (SA) and adaptive simulated annealing genetic algorithm (ASAGA) are used for selecting questions from a question bank and generating a tests. The crossover and mutation operator of standard GA can not be directly usable for generating test, since integer-coded individuals have to be used and these operators produce duplicated genomes on individuals. In order to solve this problem, a mutation operation is proposed for preventing the duplications on crossovered individuals and also directing the search randomly to the new spaces. A database containing classified test questions is created together with predefined attributes for selecting questions. A particular test can be generated automatically, without active participation of the academician. The experiments and comparative analysis show that GA with proposed mutation operator is successful as nearly 100 percent and it produces results in noteworthy computational times.

**Keywords:** Genetic algorithm, Simulated annealing, Computer based assessment, Test generating.

## 1 Introduction

Computer-based assessment is attractive for saving time with large classes. It may also serve to shift the exam burden from lecturers to others, such as computer technicians administering the system [1]. However, computer-based assessment systems have some limitations. Construction of good objective questions requires skill and practice and so is initially time consuming. Assessors and invigilators need training in assessment design, and examinations management [2]. Under certain circumstances it is desirable to have a larger number of questions than would normally be needed for a single test, for security reasons.

Many computer-based assessment tests are mainly or exclusively of the multiple-choice type [3]. The traditional descriptive answer questions are generally easier to set but it must be read and assessed by human graders. Grading of descriptive answer examinations is therefore slow, costly and suffers from the foibles of human variation in judgment and performance [4]. For a good many of students or for ongoing automated assessment, multiple-choice tests are highly advantageous.

There are many studies in the literature that are mostly based on random selection of questions from a question bank. However, either these works don't use any criteria for question selection or use some criteria but do need the participation of the academician to optimize them. If lots of questions are selected, human optimization will be failed. Assessments can be designed which draw a certain number of questions, at random, thereby producing a unique subset of questions. If no criteria are used, all of the questions in question bank are feasible and can be selected to create a unique subset. Thus, all of the questions in the subset may be difficult or easy. Moreover, frequently asked questions in the past examinations may be selected again in new subset. Even more, a question may be selected twice or more in the same test. If some criteria are used, number of feasible questions in question bank is reduced, it may even be less than the predefined number of questions of a certain test size. In a condition like this, an optimization algorithm is needed to select the most appropriate unfeasible questions to complete the number of questions of a certain test size.

In this study, heuristic optimization methods which are genetic algorithm (GA), simulated annealing (SA) and adaptive simulated annealing genetic algorithm (ASAGA) are used to optimize predefined criteria for selecting questions from a question bank. The crossover and mutation operator of standard GA can not be directly usable for generating test, since integer-coded individuals have to be used and these operators produce duplicated genomes on individuals. In order to solve this problem, a mutation operation is proposed for preventing the duplications on crossed individuals and also directing the search randomly to the new spaces. It is known from the literature that, the hybrid of GA and SA, that is ASAGA, have great success on solving some problems. In this study, moreover, the results of these three methods are compared for test generating problem. The average difficulties of questions, the number of feasible questions, the number of violated questions, the successes of generating and the computation times are observed. The experiments and analysis show that GA with proposed mutation operator is greatly successful on feasible question selection from a question bank.

## 2 Heuristic Optimization Methods

The heuristic methods are iterative search techniques that can search not only a local optimal solution but also a global optimal solution. In the heuristic methods, the techniques frequently applied to the optimization problems are GA and SA, etc. They are general-purpose search techniques based on principles inspired from the natural systems. These methods have the advantage of searching the solution space more thoroughly, and avoiding premature convergence to local minima.

### 2.1 Genetic Algorithm

GA is a parallel and global search technique that emulates natural genetic operators. Because it simultaneously evaluates many points in the parameter space, it is more likely to converge toward the global optimum. It is not necessary that the search space be differentiable or continuous. GA applies operators inspired by the mechanics of natural selection to a population of binary strings encoding the parameter space. At

each generation, it explores different areas of the search space, and then directs the search to regions where there is a high probability of finding a better solution.

The overall GA optimization system is described by the algorithm given in below. GA starts with an initial population of coded strings, which are generally called *individual* or *chromosome* and randomly selected. An individual is a potential solution of the problem and represents a set of parameters. The size of population varies from one problem to another.

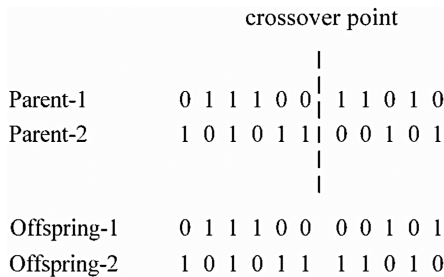
*Algorithm of GA*

```

t=0
Randomly generate initial population  $\delta(0)$ 
Until termination, Do:
    Evaluate  $\delta(t)$ 
     $\delta_1(t)=\text{Select}(\delta(t))$ 
     $\delta_2(t)=\text{Crossover}(\delta_1(t))$ 
     $\delta_3(t)=\text{Mutate}(\delta_2(t))$ 
     $\delta(t+1)=\delta_3(t)$ 
    t=t+1
end Do.
    
```

Each individual in the population is then assigned a probability of survival, in other words *fitness*, according to the objective function values of that and other individuals. To maintain uniformity over various problem domains, objective function value is rescaled to a fitness value [6]. Candidate individuals that might survive into the next generation are selected based on their fitness values. Roulette wheel selection is a commonly used method for selection. After the selection, crossover and mutation operations take place respectively. Generally, crossover combines the features of two parent chromosomes to form two offspring (as shown in Fig. 1), with the possibility that good chromosomes may generate better ones [7]. The bits of two individuals after the crossover point are swapped with a probability of crossover rate [8]. Crossover operation expands the search space around the fittest individuals [9].

Next, all candidate individuals in the population are subjected to the random mutation. This is a bit-wise binary complement operation, as shown in Fig. 2, applied uniformly to all bits of all individuals in the population with a probability of mutation



**Fig. 1.** Crossover for binary individuals



Before Mutation	0 1 1 1 0 0 0 0 1 0 1
After Mutation	0 1 1 1 0 0 1 0 1 0 1

**Fig. 2.** Mutation for binary individual

rate. The mutation operation expands the search space to regions that may not be close to the current population, thus ensuring a global search [10].

In the case of generation replacement, the individuals in the old population are replaced by the offspring. The cycle of evolution is repeated until a desired termination criterion is reached. This criterion can be set by the number of evolution cycles (computational runs) or a predefined value of objective function [11].

## 2.2 Simulated Annealing

In SA, the optimization problem is simulated as an annealing process. The natural process of optimization that takes place in a slowly cooling metal (annealing) guarantees that the structure of the metal reaches the crystal structure corresponding to the minimum energy. In this natural process, a transition from a structure corresponding to an energy level of  $E$  to that corresponding to  $E+\Delta E$  takes place, with a probability given by the Boltzman distribution function  $e^{-\Delta E/KT}$ . The above process allows a slowly cooling metal to escape from crystal structures corresponding to local minimum energy states in its search for the globally minimum energy state. In SA, the objective function to be minimized is analogous to the energy in the crystal structure and the temperature is analogous to a control parameter in the algorithm [12]. The algorithm of SA is given below [13].

### Algorithm of SA

```

begin
  iteration  $I=0$  and temperature  $TI=T_0$ ;
  generate initial new solution string  $\delta n$ ;
  current solution  $\delta c=\delta n$ ;
  cost  $F(\delta c)$ ;
  repeat
    repeat
      generate a new solution string  $\delta n$  in the neighborhood
        of  $\delta c$ ;
      if ( $F(\delta n)<F(\delta c)$ ) then  $\delta c=\delta n$  with higher probability;
      else  $\delta c=\delta n$  with lower probability;
    until(terminating condition at current temperature);
  next iteration  $I=I+1$ ;
  new temperature,  $TI=g(T_0, I)$ ;
  until (SA stopping criteria);
end.
```

### 2.3 Adaptive Simulated Annealing Genetic Algorithm

Owing to having the ability to seek for near global optimal solutions, SA has been applied to numerous optimization problems. Performance of raw SA is not satisfactory. In the original SA algorithm, a large share of the computation time is spent in randomly generating and evaluating solutions that turn out to be infeasible. Expected improvement of SA has not been done yet. Authors always try to merge SA with other methods where SA solves one of the parts of problem [13].

In order to improve GA, SA can be merged with GA. The hybrid algorithm ASAGA preserves the merits of GA by changing the only mutation operator of GA with SA [14, 15]. as given in algorithm below. The new mutation operates like SA as follows. It generates a random individual adjacent the individual generated by crossover operator of GA. It accepts if the new individual is better than the original individual. Otherwise, the new individual is accepted according to some probability.

#### Algorithm of ASAGA

```

Iteration i=0 and temperature TI=T0;
Randomly generate initial population  $\delta(0)$ 
Until termination, Do:
  Evaluate  $\delta(i)$ 
   $\delta_1(i)$ =Select( $\delta(i)$ )
   $\delta_2(i)$ =Crossover( $\delta_1(i)$ )
  repeat
    generate a new solution string  $\delta n$  in the neighborhood
      of  $\delta_2(i)$ ;
    if ( $F(\delta n) < F(\delta_2(i))$ ) then  $\delta_3(i) = \delta n$  with higher
      probability;
    else  $\delta_3(i) = \delta_1(i)$  with lower probability;
  until(terminating condition at current temperature);
  new temperature,  $TI = g(T0, i)$ ;
   $\delta(i+1) = \delta_3(i)$ 
   $i = i + 1$ 
end Do.

```

## 3 Application of Algorithms to Test Generation Problem

### 3.1 The Structure of the Individual

In much early GA work the individuals were binary-coded, but more generally individuals may be binary, integer or decimal-coded, and may also take matrix form. In a test preparation problem, it is possible to use binary-coded individuals. However in this case, the individual size would be greater and conversion between binary and integer individuals take much computational time. Therefore, in this study, the question numbers to be selected are represented by an integer-coded individual:

$$\mathbf{x} = \{Question_1, Question_2, \dots, Question_M\} \quad (1)$$

where  $m$  is the predefined test size and  $\mathbf{x} \in D^m$  is the vector of decision variables. In Fig. 3, the population with  $n$  individuals is shown.

	Q1	Q2													Qm
individual-1	78	43	56	12	23	182	223	236	7	14	67	142	171	36	163
individual-2	21	78	231	179	152	73	98	102	16	37	55	144	193	7	221
individual-n	191	17	163	121	3	58	93	11	159	43	102	182	134	88	205

Fig. 3.  $n*m$  integer-coded population

### 3.2 The Constrained Objective Function

The penalty function method, in which the objective function value is degraded by some function of constraint violation, has been the most popular for constrained optimization by GA [16]. A penalty function defines the fitness value of an infeasible individual [17].

In the present study, the idea is to apply a set of criteria to decide the selection process as follows:

- Any feasible solution is preferred to any infeasible solution.
- Between two feasible solutions, any of them is preferred.
- Between two infeasible solutions, the one having smaller constraint violation is preferred.

Based on these criteria, the objective function of the constrained optimization is

$$F(\mathbf{x}) = f(\mathbf{x}) + \sum_1^m viol(\mathbf{x}) \tag{2}$$

where  $f(\mathbf{x})$  is the objective function value, and  $viol(\mathbf{x})$  the summation of all the violated constraints, such that  $viol(\mathbf{x})=0$  if  $\mathbf{x}$  is feasible and  $viol(\mathbf{x})>0$  otherwise. The constraints are:

1. Frequently selected questions in previous tests are unfeasible. A previously unselected question violation is equal to 0 and it is feasible; violation of a question selected once is equal to 1 and so on.
2. Difference between the difficulty of selected question and the requested difficulty level must be zero or minimum.

### 3.3 The Combination of Crossover and Mutation Operators

During the crossover step of the algorithm, segments are cut-and-spliced between individuals. In the test generating problems, since the integer-coded individuals are used, crossover frequently generates illegal offsprings. For example, if two parents are crosseovered as shown in Fig. 4, the produced offsprings are clearly illegal, since each of them have duplicated question numbers.

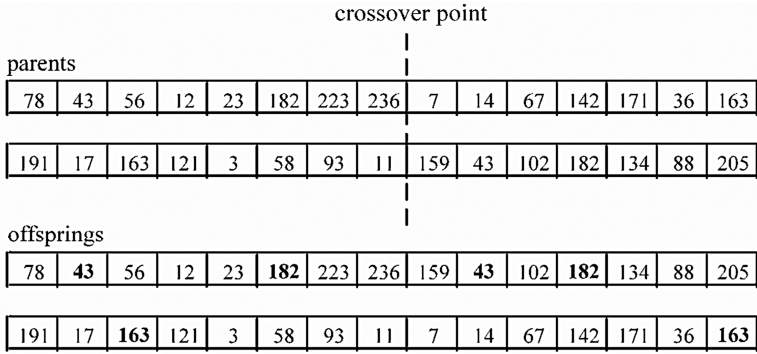


Fig. 4. Duplication problem after crossover for integer-coded individual

In Fig. 5, the flowchart of proposed method is shown. It must be again said that the mutation operation should expand the search space to the regions that may not be close to the current population, in order to ensure a global search. If one of the duplicated parameters in illegal offspring is replaced with a randomly selected one, which is not member of the illegal offspring, both the duplication problem is solved and the search space can be expanded to new regions. Such a mutation operation seems like a part of crossover operation and the mutation operation itself.

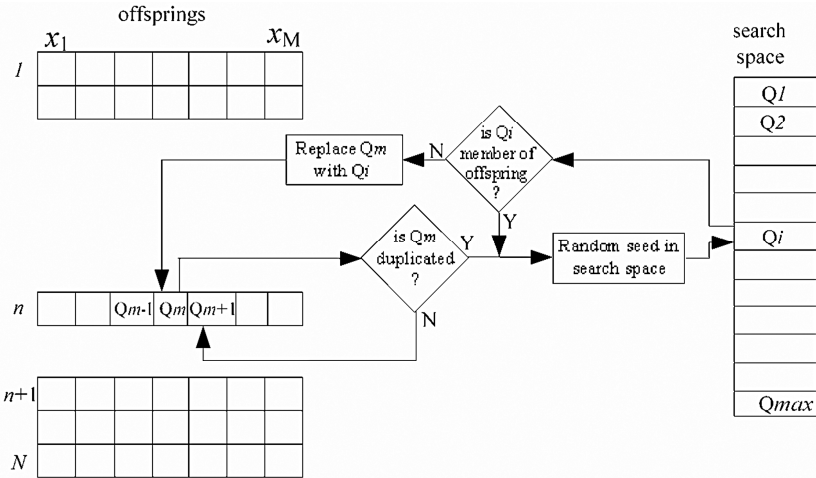


Fig. 5. Flowchart of duplication prevention with mutation

### 4 Experiments and Analysis

A question bank database, which has 240 questions, was created firstly for generating test. Each question is characterized with some scalar attributes, such as question

number, difficulty of the question, chapter and sub-title number that the question belongs to, and history of appearance on previously generated tests. It is necessary that the difficulty levels of questions are predefined in the question bank. In practice, the decision of which difficulty level a question belongs to is not an easy task. Difficulty level of a question is determined once by the instructor when the question is created into question bank database. After that, it is updated according to statistical data about students' performance on each question. History of appearance field on the question bank database is modified when generating a new test. This field is incremented by 1 for all questions in the new generated test. This gives less chance to these questions than unselected questions for the next generating of a new test.

Since it is convenient for representation of question number, integer-coded individuals were used. In order to select the fittest individuals for moving them to the next generation, roulette-wheel selection mechanism was used. The best individual was directly copied into the next generation by using elitism. Effects of population

**Table 1.** Results of generated tests by GA

Test size	Requested difficulty	Size of feasible subset	Results of generated test by GA				Success (%)	Running Time (sec.)
			Average difficulty	Feasible questions	History violations	Difficulty violations		
20	1	24	1	20	0	0	100.0	9.69
	2	72	2	20	0	0	100.0	2.03
	3	120	3	20	0	0	100.0	1.84
	4	72	4	20	0	0	100.0	2.03
	5	24	5	20	0	0	100.0	7.94
24	1	24	1	24	0	0	100.0	19.34
	2	72	2	24	0	0	100.0	3.73
	3	120	3	24	0	0	100.0	3.26
	4	72	4	24	0	0	100.0	3.14
	5	24	5	24	0	0	100.0	17.94
40	1	24	1.22	24	7	9	100.0	50.24
	2	72	2	40	0	0	100.0	19.65
	3	120	3	40	0	0	100.0	10.19
	4	72	4	40	0	0	100.0	24.87
	5	24	4.83	24	9	7	100.0	51.98
50	1	24	1.26	24	13	13	100.0	100.91
	2	72	2	50	0	0	100.0	32.91
	3	120	3	50	0	0	100.0	16.27
	4	72	4	50	0	0	100.0	55.70
	5	24	4.76	24	14	12	100.0	120.80
70	1	24	1.31	24	24	22	100.0	361.06
	2	72	2	68	2	0	97.1	327.73
	3	120	3	70	0	0	100.0	59.25
	4	72	4	68	2	0	97.1	266.63
	5	24	4.68	24	24	22	100.0	319.09

size, crossover method and maximum generation number were investigated and crossover method was taken as one-point, the population size was taken as 100, mutation rate is taken as 0.02 and the maximum generation number was taken as 1000 for GA and ASAGA, and 100000 for SA.

By using GA, SA and ASAGA, twenty five tests were generated for each method. While generating the tests, five certain numbers of test sizes and five difficulty levels were requested. The results of generated tests are given in Table 1 for GA, Table 2 for SA and Table 3 for ASAGA. In the tables, the requested test sizes and difficulties, and the number of feasible questions in the question bank according to requested values are shown. In addition, the average difficulties of generated tests, the number of feasible questions in each generated test, the number of violated questions that historically appeared in previous tests, the number of violated questions that have different difficulty from the requested difficulty in average. Finally, the table shows the feasible question selection successes and the computation times of the methods.

**Table 2.** Results of generated tests by SA

Test size	Requested difficulty	Size of feasible subset	Results of generated test by SA				Success (%)	Running Time (sec.)
			Average difficulty	Feasible questions	History violations	Difficulty violations		
20	1	24	2	8	6	20	33.3	46.17
	2	72	2	15	5	0	20.8	71.40
	3	120	3	17	3	0	14.1	107.93
	4	72	3.9	9	9	2	12,5	67.45
	5	24	4	7	5	20	29.1	44.87
24	1	24	2.04	5	11	25	20.8	105.85
	2	72	2.13	12	8	3	16.6	479.54
	3	120	3	19	5	0	15.8	107.25
	4	72	3.92	13	10	2	18.0	74.03
	5	24	3.88	6	8	27	25.0	55.04
40	1	24	2.28	7	18	51	29.1	88.46
	2	72	2.28	21	14	11	29.1	38.68
	3	120	3	31	9	0	25.8	82.03
	4	72	3.77	17	19	9	23.6	40.37
	5	24	3.65	4	17	54	16.6	0.70
50	1	24	2.42	8	20	71	33.3	67.53
	2	72	2.36	21	25	18	29.1	45.26
	3	120	3	35	15	0	29.1	54.28
	4	72	3.68	24	19	16	33.3	42.39
	5	24	3.64	6	21	68	25.0	82.43
70	1	24	2.46	11	31	102	45.8	43.71
	2	72	2.57	29	28	40	40.2	37.31
	3	120	3.01	44	25	1	36.6	123.17
	4	72	3.48	28	32	36	38.8	51.21
	5	24	3.41	9	30	111	37.5	42.26

**Table 3.** Results of generated tests by ASAGA

Test size	Requested difficulty	Size of feasible subset	Results of generated test by ASAGA				Success (%)	Running Time (sec.)
			Average difficulty	Feasible questions	History violations	Difficulty violations		
20	1	24	1	17	3	0	70.8	9.37
	2	72	2	17	0	3	23.6	3.28
	3	120	3	20	0	0	16.6	1.33
	4	72	4	20	0	0	27.7	2.79
	5	24	4.94	16	3	1	66.6	8.15
24	1	24	1.04	12	7	1	50.0	15.33
	2	72	2	18	0	2	25.0	10.52
	3	120	3	20	0	0	16.6	6.09
	4	72	4	18	0	2	25.0	8.86
	5	24	4.95	13	6	1	54.1	13.11
40	1	24	1.17	23	12	7	95.8	97.92
	2	72	2	37	0	3	51.3	29.02
	3	120	3	40	0	0	33.3	13.78
	4	72	4	38	0	2	52.7	27.56
	5	24	4.70	23	6	12	95.8	102.32
50	1	24	1.2	23	18	10	95.8	180.02
	2	72	2	46	0	4	63.8	110.78
	3	120	3	50	0	0	41.6	74.53
	4	72	4	46	0	4	63.8	101.41
	5	24	4.82	23	19	11	95.8	169.84
70	1	24	1.47	24	25	34	100.0	532.71
	2	72	2	62	3	0	86.1	294.32
	3	120	3	69	1	0	57.5	384.50
	4	72	4	59	4	2	81.9	640.37
	5	24	4.54	24	26	32	100.0	288.03

When the requested test size is less than the number of feasible questions, GA is able to select all feasible questions easily. If it is greater than the number of feasible questions, GA firstly selects the all feasible ones and then the least violated ones to complete test size. For example; when 20 questions and 1 difficulty level is requested, GA selects the questions all of which are feasible, have the difficulty of 1, and which have not appeared in previous tests. When 40 questions and 1 difficulty level is requested, GA selects 24 feasible questions, 7 previously appeared questions, and 9 questions with different difficulty. In this generating, GA is able to select all of the feasible questions firstly, and then selects the least violated questions to complete 40 questions. This generating has 100% success, since GA is able to select the most appropriate questions.

The computation times, at which the solutions are obtained by the three methods, are support the comment given above. For all test sizes, GA finds a solution in less

time when the requested test size is less than the feasible question size (see the difficulty of 2, 3, and 4 for all test sizes). However, as the requested test size grows computation time of solution increases, since the individual size used in GA is extended.

GA is the most, ASAGA is the second and SA is the least successful method when the feasible question selection successes of them are compared. GA is able to select all of the feasible questions in 23 cases and nearly all questions in 2 cases. SA does not pass over the 45.8 success rate.

GA has less computation time than ASAGA, in general. In the case of ASAGA has the less computation time than GA, this time it is less successful than GA. It is difficult to compare SA with GA and ASAGA, since SA has unsystematic computation time results.

The all results and comparisons show that GA with proposed mutation operator is greatly successful on feasible question selection form a question bank. SA seems unsuccessful, because it depends on neighbourhood and the adjacent questions in question bank or database have no common attributes. This means that feasible questions are scattered to all search space.

## 5 Conclusion

In this study, heuristic optimization methods which are GA, SA and ASAGA are used to optimize predefined criteria for selecting questions from the question bank. The crossover and mutation operator of standard GA can not be directly usable for generating test, since integer-coded individuals have to be used and these operators produce duplicated genomes on individuals. In order to solve this problem, a mutation operation is proposed for preventing the duplications on crossed individuals and also directing the search randomly to the new spaces.

A database containing classified test questions was created together with predefined attributes for selecting questions. By using GA, SA and ASAGA, twenty five tests were generated for each method. In the generatings, five certain numbers of test sizes and five difficulty levels were requested. The average difficulties of questions, the number of feasible questions, the number of violated questions, the successes of generating and the computation times are observed.

An other result of the study is that, when the requested test size is less than the number of feasible questions, GA is able to select all feasible questions easily. If it is greater than the number of feasible questions, GA firstly selects the all feasible ones and then the least violated ones to complete test size.

One more result is that, for all test sizes, GA finds a solution in less generations when the requested test size is less than the feasible question size. However, as the requested test size grows generation number of solution increases, since the individual size used in GA is extended.

GA is the most, ASAGA is the second and SA is the least successful method when the feasible question selection successes of them are compared. GA has less computation time than ASAGA, in general. It is difficult to compare SA with GA and ASAGA, since SA has unsystematic computation time results.



The experiments and comparative analysis show that GA with proposed mutation operator is successful as nearly 100 percent and it produces results in noteworthy computational times.

## References

1. Thelwall, M.: Computer-Based Assessment: a Versatile Educational Tool. *Computers & Education* 34, 37–49 (2000)
2. Fei, T., Hag, W.J., Toh, K.C., Qi, T.: Question Classification for E-learning by Artificial Neural Network. In: *Proceedings of ICICS-FCM ZW3*, Singapore, pp. 1757–1761 (2003)
3. Protia J., Bojia D., Tartalja I.: Test: Tools for Evaluation of Students Tests - a Development Experience. In: *Proceedings of 31st ASEE/IEEE Frontiers in Education Conference*, Reno, NV, F3A6-12 (2001)
4. Brown, R.W.: Multi-choice Versus Descriptive Examinations. In: *Proceedings of 31st ASEE/IEEE Frontiers in Education Conference*, Reno, NV, T3A13-18 (2001)
5. Prabhu, D., Buckles, B.P., Petry, F.E.: Genetic Algorithms for Scene Interpretation from Prototypical Semantic Description. *Int. J. Intel. Syst.* 15, 901–918 (2000)
6. Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms: Concepts and Applications. *IEEE Trans. Ind. Electron.* 43, 519–533 (1996)
7. Herrera, F., Lozano, M., Sánchez, A.M.: A Taxonomy for the Crossover Operator for Real-Coded Genetic Algorithms: an Experimental Study. *Int. J. Intel. Syst.* 18, 309–338 (2003)
8. Spears, W.M., Anand, V.: A Study of Crossover Operators in Genetic Programming. In: *Proceedings of 6th International Symposium on Methodologies for Intelligent Systems*, pp. 409–418. Springer, Heidelberg (1991)
9. Kristinsson, K., Dumont, G.A.: System Identification and Control Using Genetic Algorithms. *IEEE Trans. Syst. Man. Cybern.* 22, 1033–1046 (1992)
10. Yildirim, M., Erkan, K.: Determination of Acceptable Operating Cost Level of Nuclear Energy for Turkey's Power System. *Energy* 32, 128–136 (2007)
11. Zhu, F., Guan, S.U.: Ordered Incremental Training with Genetic Algorithms. *Int. J. Intel. Syst.* 19, 1239–1256 (2004)
12. Annakkage, U.D., Numnonda, T., Pahalawaththa, N.C.: Unit Commitment by Parallel Simulated Annealing. *IEE Proc.-Gener. Transm. Distrib.* 142(6), 595–600 (1995)
13. Senjyu, T., Saber, A.Y., Miyagi, T., Urasakin: Absolutely Stochastic Simulated Annealing Approach to Large Scale Unit Commitment Problem. *Electric Power Components and Systems* 34, 619–637 (2006)
14. Jeong, I.K., Lee, J.J.: Adaptive Simulated Annealing Genetic Algorithm for System Identification. *Engng. Applic. Artif. Intell.* 9(5), 523–532 (1996)
15. Yildirim, M., Erkan, K., Ozturk, S.: Power Generation Expansion Planning with Adaptive Simulated Annealing Genetic Algorithm. *Int. J. Energy Res.* 30, 1188–1199 (2006)
16. Kumar, N., Shanker, K.: A Genetic Algorithm for FMS Part Type Selection and Machine Loading. *Int. J. Prod. Res.* 38, 3861–3887 (2000)
17. Lemonge, A.C.C., Barbosa, H.J.C.: An Adaptive Penalty Scheme for Genetic Algorithms in Structural Optimization. *Int. J. Numer. Meth. Engng.* 59, 703–736 (2004)

# Author Index

- Acosta Guadarrama, J.C. 260  
Aguirre, Eugenio 747, 789  
Aguirre, José Luis 381  
Akramifar, Seyed Ali 1067  
Alcaide, David 821  
Alimi, Adel M. 623  
Alique, J.R. 1162  
Almanza, Dora L. 650  
Almazán-Delfín, Ana J. 758  
Alonso, César L. 316  
Alonso, David 316  
Aragón, Victoria S. 19  
Arredondo V., Tomás 811  
Arriola, Veronica E. 725  
Arroyo, Gustavo 452  
Astengo-Noguez, Carlos 52  
Aviña-Cervantes, J. Gabriel 650  
Aviña, Gabriel 161
- Baccour, Leila 623  
Bangham, J. Andrew 580  
Barrera-Cortes, Josefina 1184  
Barriga Rodriguez, Leonardo 705  
Baruch, Ieroham 1184  
Batyrshin, Ildar 9  
Bello, Rafael 483  
Berrones, Arturo 94  
Biajoli, Fabrício Lacerda 83  
Bonev, Boyan 431  
Borja Macías, Verónica 225  
Borji, Ali 61  
Botello, Salvador 118  
Bouamrane, Karim 139
- Callau, Mar 316  
Calvo, Hiram 912  
Camacho Nieto, Oscar 9  
Carbonaro, Antonella 1195  
Carvalho, Ariadne Maria Brito Rizzoni 966  
Casadei, Giorgio 1195  
Casas, Gladys 441  
Castelán, Mario 640, 758  
Castilla Valdez, Guadalupe 1078
- Cazorla, Miguel Angel 431  
Cellier, François E. 1173  
Chao, Wen-Han 955  
Chávez-Aragón, Alberto 612  
Chávez, María del Carmen 441  
Chen, Changxiong 933  
Chen, Dezhi 550  
Chen, Yue-Xin 955  
Clark, Jonathan H. 839  
Coello Coello, Carlos A. 19, 30, 41, 128  
Costa, Hugo 944  
Costa, João C. Weyl 496  
Costa, Rui P. 944  
Cruz-Barbosa, Raúl 472  
Cruz Reyes, Laura 1078  
Cruz Sánchez, Vianey Guadalupe 590  
Cruz-Vega, Israel 1184  
Cuevas-Tello, Juan C. 559
- da Cunha, Iria 872  
Dang, Chuangyin 1  
de-la-Rosa-Vazquez, Jose M. 150  
Di Persia, Leandro E. 518  
Dehmer, Matthias 540  
Delbem, Alexandre C.B. 72  
Delgado-Orta, J. Francisco 1078  
Denkowski, Michael 1025  
Devy, Michel 650  
Didi Biha, Mohamed idxquad 883  
Dou, Huiming 550
- El-Bèze, Marc 985  
Emmert-Streib, Frank 540  
Escobet, Antoni 1173  
Escolano, Francisco 431  
Esquivel, Susana C. 19
- Falcón, Rafael 441, 483  
Federson, Fernando M. 72  
Fernández, Silvia 861, 872  
Flores Romero, Juan J. 30  
Francês, Carlos Renato L. 496  
Freund, Wolfgang 811  
Frolov, Alexander 671  
Fuentes Cabrera, Juan C. 41

- Galicia-Haro, Sofia N. 922  
Gallegos-Funes, Francisco J. 150, 660  
García, Guadalupe 161  
García-Silvente, Miguel 747, 789  
García, Zenaida 1206  
Garro, Beatriz A. 694  
Garza, Luis E. 1162  
Garza, Sara E. 381  
Garza-Domínguez, Ramiro 1122  
Geibel, Peter 203, 409, 850  
Gelbukh, Alexander 215, 922  
Gervás, Pablo 944  
Ghassem-Sani, Gholamreza 1056, 1067  
Goddard, John 1004  
Gomez, Jonatan 462  
Gómez, Moises 789  
Gómez, Octavio 420  
González, Guillermo 215  
González, Jesús A. 420  
González B., J. 1078  
Gorrostieta Hurtado, Efen 705  
Grau, Ricardo 441  
Gust, Helmar 203, 850  
Guzmán-Cabrera, Rafael 831  
Guzmán, Enrique 601
- Haenni, Rolf 236, 248  
Halbritter, Florian 409  
Hamdadou, Djamilia 139  
Hannon, Charles J. 839, 1025  
Hayet, Jean-Bernard 736, 800  
Hernández, Arturo 118  
Hernández, Donato 161  
Hernández-Gutierrez, Andres 650  
Hernández Zavala, Antonio 9  
Herrera Ortiz, Juan A. 1078  
Hervás, Raquel 944  
Houshmandan, Amirali 1056  
Hu, Daiping 550  
Hung, Chih-Cheng 570  
Húsek, Dušan 671
- Ibarra, Mario A. 650  
Imbert, Ricardo 370
- Jeon, Gwanggil 483  
Jeong, Jechang 483  
Jiang, Peilin 1046  
Juárez Gambino, Omar 912  
Juarez G., Roberto 1089
- Kaba, Bangaly 883  
Kazienko, Przemyslaw 529  
Kessler, Rémy 985  
Khabou, Mohamed A. 623  
Klempous, Ryszard 1132  
Kozareva, Zornitsa 996  
Krumnack, Ulf 203  
Kühnberger, Kai-Uwe 203, 850  
Kuri-Morales, Ángel 399  
Kuroiwa, Shingo 1035, 1046
- Ledesma, Sergio 161  
Lee, Na-Young 1143  
Leon, Elizabeth 462  
León, Maikel 1206  
Levner, Eugene 821  
Li, Zhou-Jun 955  
Liang, Jiye 1  
Liu, Chuanhan 894  
Liu, Hui 933, 975  
Lizárraga, Giovanni 118  
Lorena, Luiz Antonio Nogueira 83, 1099  
Lu, Ruzhan 933, 975
- Madarasmi, Suthep 683  
Magatão, Leandro 1110  
Maghrebi, Wafa 623  
Malec, Jacek 359  
Mariaca-Gaspar, Carlos-Roman 1184  
Martínez, César E. 1004  
Martínez, Natalia 1206  
Martínez-Jiménez, Leonardo 650  
Matsumoto, Kazuyuki 1035  
Medina, Jesús 271  
Mejía, L. Felipe 634  
Mejía-Guevara, Iván 399  
Mejía-Lavalle, Manuel 452  
Melo, Vinícius V. de 72  
Meurs, Marie-Jean 883  
Milone, Diego H. 518, 1004  
Montaña, José Luis 316  
Montes-y-Gómez, Manuel 831, 904  
Montoyo, Andrés 996  
Montúfar-Chaveznavia, Rodrigo 713  
Morales, Eduardo F. 420, 452  
Morales-Menendez, Ruben 1162  
Morán L., Luis E. 779  
Moravec, Pavel 671  
Moreira, Jorge E. 441  
Moreno-Escobar, Jose A. 150

- Mu, Xiangyang 392  
 Muñoz, César 811  
 Muñoz-Salinas, Rafael 747, 789  
 Murrieta-Cid, Rafael 800  
 Musiał, Katarzyna 529  
  
 Nagano, Marcelo Seido 1099  
 Nakano-Miyatake, Mariko 769  
 Nakhost, Hootan 1056  
 Nebot, Ángela 1173  
 Nieto-Yáñez, Diana M. 1078  
 Nieves, Juan Carlos 294  
 Novoa, Elizabeth 327  
 Nowaczyk, Sławomir 359  
  
 Ojeda-Aciego, Manuel 271  
 Olivares-Mercado, Jesus 769  
 Oropeza Rodríguez, José Luis 1015  
 Ortega-Mendoza, Rosa M. 904  
 Osorio, Mauricio 283, 294  
 Osorio L., Maria A. 1089  
 Ostrowski, Richard 105  
 Ovchinnikova, Ekaterina 203  
  
 Paris, Lionel 105  
 Pastrana Palma, Alberto 580  
 Paúl, Rui 747  
 Pedraza Ortega, Jesus Carlos 634, 705  
 Peña, Dexmont 94  
 Peña-Kaltekis, Juan 462  
 Pereira, Francisco C. 944  
 Perez-Meana, Hector 769  
 Pérez-Meza, Mónica 713  
 Piater, Justus 736  
 Pinto, David 821  
 Pinto Elías, Raúl 779  
 Pinto Júnior, Dorival L. 72  
 Pogrebnyak, Oleksiy 601  
 Polyakov, Pavel 671  
 Ponomaryov, Volodymyr 150, 660  
 Pouly, Marc 236, 248  
 Pozos Parra, Pilar 225  
  
 Qian, Yuhua 1  
 Queiroz, João 370  
 Quirós, Fernando 811  
 Quiroz-Gutiérrez, Antonio 1122  
  
 Radlak, Marcin 1132  
 Ramírez, Miguel 1162  
  
 Ramírez-Sosa Morán, Marco I. 640, 758  
 Ramos Arreguin, Juan Manuel 705  
 Rangel-Valdez, Nelson 1078  
 Ren, Fuji 1035, 1046  
 Řezanková, Hana 671  
 Ribeiro Filho, Geraldo 1099  
 Riccucci, Simone 1195  
 Rivas, Angel 705  
 Rodrigues, Luiz C.A. 1110  
 Rodriguez Moreno, Jose Wilfrido 705  
 Román-Godínez, Israel 193  
 Romero-Leon, Nestor 462  
 Rosales-Silva, Alberto 660  
 Rosso, Paolo 821, 831  
 Rufiner, Hugo L. 1004  
 Ruiz-Calviño, Jorge 271  
  
 Saatchi, Sara 570  
 Saïs, Lakhdar 105  
 Salgado Jimenez, Tomas 705  
 Sanchez, Antonio 1025  
 Sánchez, Ricardo 94  
 Sánchez-Ante, Gildardo 52  
 Sánchez L., Abraham 1089  
 Sanchez-Perez, Gabriel 769  
 SanJuan, Eric 861, 872, 883  
 Santana, Ádamo L. de 496  
 Santos, Denis Neves de Arruda 966  
 Santoyo, Alejandro 634  
 Santoyo, Joaquín 634  
 Savage, Jesus 725  
 Schmidt, C.T.A. 348  
 Schütze, Oliver 128  
 Schwering, Angela 203  
 Serrato Paniagua, Ramiro 30  
 Sharma, R.R.K. 821  
 Siegel, Pierre 105  
 Snášel, Václav 671  
 Sossa, Humberto 694  
 Spinola de Freitas, Jackeline 370  
 Starostenko, Oleg 612  
 Štěpánek, Petr 305  
 Suárez Guerra, Sergio 1015  
 Sun, Tengda 337  
  
 Talbi, El-Ghazali 128  
 Tenbergen, Bastian 182  
 Tiño, Peter 172  
 Tipwai, Preeyakorn 683

- Torres-Méndez, Luz A. 640, 758  
Torres, Miguel 161  
Torres-Moreno, Juan Manuel 861, 872,  
985  
Tsuchiya, Seiji 1035
- Vallejo, Antonio Jr. 1162  
Vázquez, Roberto A. 694  
Vázquez, Sonia 996  
Velázquez Morales, Patricia 872  
Vellido, Alfredo 472  
Vergara Villegas, Osslan Osiris 590  
Villa Vargas, Luis 9  
Villaseñor-Pineda, Luis 831, 904  
Vivaldi, Jorge 872  
Vyskočil, Jiří 305
- Wachter, Michael 236, 248  
Wandmacher, Tonio 203
- Wang, Jinfeng 337  
Wu, Ruiming 550
- Xiang, Hua 1046
- Yañez, Cornelio 601  
Yañez-Márquez, Cornelio 193  
Yasuda, Gen'ichi 1151  
Yildirim, Mehmet 1218  
Yu, Chien-Chih 507
- Zapata, Carlos Mario 215  
Zepeda, Claudia 283  
Zhang, Taiyi 392  
Zhang, Xia 1  
Zhang, Yumeng 894  
Zhao, Jinglei 933, 975  
Zheng, Nanning 1046  
Zhou, Yatong 392